A Novel Approach for Performance Assessment of Human-Robotic Interaction

by

Jamil Akram Abou Saleh

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2012

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Robots have always been touted as powerful tools that could be used effectively in a number of applications ranging from automation to human-robot interaction. In order for such systems to operate adequately and safely in the real world, they must be able to perceive, and must have abilities of reasoning up to a certain level. Toward this end, performance evaluation metrics are used as important measures. This research work is intended to be a further step toward identifying common metrics for task-oriented human-robot interaction. We believe that within the context of human-robot interaction systems, both humans' and robots' actions and interactions (jointly and independently) can significantly affect the quality of the accomplished task. As such, our goal becomes that of providing a foundation upon which we can assess how well the human and the robot perform as a team. Thus, we propose a generic performance metric to assess the performance of the human-robot team, where one or more robots are involved. Sequential and parallel robot cooperation schemes with varying levels of task dependency are considered, and the proposed performance metric is augmented and extended to accommodate such scenarios. This is supported by some intuitively derived mathematical models and some advanced numerical simulations. To efficiently model such a metric, we propose a two-level fuzzy temporal model to evaluate and estimate the human trust in automation, while collaborating and interacting with robots and machines to complete some tasks. Trust modelling is critical, as it directly influences the interaction time that should be directly and indirectly dedicated toward interacting with the robot. Another fuzzy temporal model is also presented to evaluate the human reliability during interaction time. A significant amount of research work stipulates that system failures are due almost equally to humans as to machines, and therefore, assessing this factor in human-robot interaction systems is crucial. The proposed framework is based on the most recent research work in the areas of human-machine interaction and performance evaluation metrics. The fuzzy knowledge bases are further updated by implementing an application robotic platform where robots and users interact via semi-natural language to achieve tasks with varying levels of complexity and completion rates. User feedback is recorded and used to tune the knowledge base where needed. This work intends to serve as a foundation for further quantitative research to evaluate the performance of the human-robot teams in achievement of collective tasks.

Acknowledgments

Foremost, I would like to express my sincere gratitude to my advisor Dr. Fakhri Karray, for the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

I would also like to acknowledge Michael Morckos, and Sepideh Seifzadeh for their assistance and help in the CPAMI lab.

Many thanks are also due to my thesis readers and reviewers, Dr. Ramadan A. El Shatshat, Dr. Dana Kulic, and Dr. William W. Melek, as well as my external examiner Dr. Jason Gu for taking the time to assess my research work, as well as for their pertinent advice and feedback.

Dedication

This thesis is dedicated to my parents, my father *Akram Abou Saleh*, and my mother *Violette Taha Abou Saleh*, for their love, endless support and encouragement. They taught me the value of education, and that even the largest task can be accomplished if it is done one step at a time. They have never failed to give us all the financial and moral support, and sacrificed a good many years of their lives to make this work possible. They have been and will always be my great source of motivation and inspiration.

Thank you mom and dad for being with me in every step of the way, for praying for me during times of weakness, and for rejoicing with me over every little triumph. I am forever indebted to you for your understanding, endless patience, encouragement, and unwavering faith in me. I owe everything I have achieved or will achieve to you.

My sister, brother, uncles, aunts, cousins, and friends, deserve my wholehearted thanks and appreciation as well. Without your support, encouragement, and motivation, this work would have never come this far.

Thank you all.

Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

ABNF	Augmented Backus-Naur Form
ACTS	Advanced Colour Tracking System
ARCOS	Advanced Robotics Control Operating System
ARIA	Advanced Robotics Interface for Applications
ARNL	Autonomous Robotics Navigation and Localization
ASD	Autism Spectrum Disorder
CA	Context Awareness
CDRH	Center for Devices and Radiological Health
CWW	Computing with Words
DIT	Direct Interaction Time
EB	External Burden
FC	Fault Cruciality
FDA	Food and Drug Administration
FF	Fault Frequency
FFSA	Finite Fuzzy State Automata
FFSM	Finite Fuzzy State Machine

FO	Fan-Out
FR	Fault Recovery
FS	Fault Size
FSA	Finite State Automata
FSM	Finite State Machine
FT	Free Time
GMP	Generalized Modus Ponens
GPS	Global Positioning System
HA	Human Awareness
HR	Human Reliability
HRI	Human-Robot Interaction
IA	Input Alphabet
IE	Interaction Effort
IIT	Indirect Interaction Time
IT	Interaction Time
LMS	Laser Measurement Sensor
MA	Machine Awareness
MCL	Monte-Carlo Localization
MW	Mental Workload
NS	Number of Subtasks
NT	Neglect Tolerance
OA	Output Alphabet
PCM	Pulse-Code Modulation
RAD	Robot Attention Demand

RM Resource Manager

SDK Software Development Kit

TC Task Completion

TE Task Effectiveness

Tr Human Trust in Automation

TS Task Sophistication

UA Unmanned Aircraft

USAR Urban Search and Rescue

VASRE Vestec Automatic Speech Recognition Engine

WI Word Interpreter

XML Extensible Markup Language

Chapter 1

Introduction

Human-robot interaction is the interdisciplinary study of interaction dynamics between humans and robots. It involves such disciplines as: artificial intelligence, natural language understanding, psychology, communication, and ethics. It addresses how humans interact with robots, and how best to design robotic systems that are capable of accomplishing interactive tasks in human environments safely and effectively [4].

Human-robot interaction has recently been receiving considerable attention due to the rapid advances in the field of robotics - advances focused on endowing robots with higher-level cognitive capabilities. These capabilities include the ability to reason, act, and perceive in dynamic, partially known, and unpredictable environments in a robust manner, by naturally interacting with humans. This allows robots to perform complex tasks within highly uncertain environments, such as in the area of urban search and rescue [5] [6], assistive robotics [7] [8], and police and military [9], to name a few.

Many systems have been implemented toward achieving this goal, but run the risk of being ignored if appropriate benchmarking procedures - allowing comparing the actual practical results with reference to standard accepted procedures - are not in place. Therefore, developing a generalized set of metrics that assess the performance of the human-robot system becomes crucial.

For several years, and in many technical fields, a wide range of performance metrics were used by the research community. Such metrics lacked generalizability due to a bias toward application-specific measures. Each metric was addressed to satisfy a specific application's needs, and were thus incomparable. More attention was then devoted to the core questions of the field in order to develop a common set of performance metrics; the goal became to present a set of common metrics that can assess the performance of human-robot teams. However, the incredibly diverse range of human-robot applications significantly increased the difficulty in defining such common metrics. Therefore, it may not be feasible to identify metrics that can accommodate the entire application space, as most metrics do not translate well between domains or even sub-domains [10]. And hence, it may be necessary to rely on measures that at least provide the benefits afforded by familiar methods and scoring, even when they don't ensure comparability across the whole application space.

The best-known performance metrics presented in the literature are those that measure task effectiveness (TE) [11], [12]. A TE metric is some measure of how well a task is actually performed. Such metrics can be classified as [13]:

- **time-based** metrics that measure the speed of performance, or the time needed for a successful completion of a specific task.
- **error-based** metrics that attempt to estimate or measure the number of mistakes or damage occurred while completing a task.
- **coverage-based** metrics that estimate how much of some larger goal is achieved.

Situation awareness also finds itself as another emerging metric used in the literature to assess system performance. Situation awareness is a field of research that commonly examines the information requirements of humans for special jobs such as facility monitoring or flying aircraft [14]. Endsley [15] defines situation awareness as "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future".

Many other metrics have been presented in the literature, however most of them are domain-specific measures. Examples of such metrics are the ones addressed to assess the performance of assistive robotics systems, such as systems that help people with autism spectrum disorder, provide elders with care and assistance, as well as other assistive systems such as intelligent wheelchairs, assistive robotic arms, and external limb prostheses [16].

1.1 Motivation

The field of performance evaluation metrics has been capturing a lot of recent attention, especially with the fast growth in the fields of cognitive robotics and human-robot interaction systems. This work is motivated by:

- the emergence of many systems that are implemented toward achieving higher order functions through learning, development, and human-machine interaction, which run the risk of being ignored, if appropriate benchmarking procedures are not in place. Therefore, presenting a generalized metric framework that assesses the performance of the human-robot system becomes crucial.

- the lack of a generalized set of performance metrics that can span much of the robotics and human-robot interaction application space, where most presented metrics are biased toward some specific application design, and do not translate well to other applications.

- the lack of performance evaluation measures that assess the performance of the human and the robot as a team. Typically, research that focuses on performance assessment of systems integrating human and robot tends to disregard the capability of one of the agents. Although research in human-robot performance assessment is expanding, approaches that integrate the contributions of both human and robot agents have been minimally addressed. We attempt to address these limitations by developing a systematic approach to assess system performance of human-robot systems in achievement of collective tasks. In this work, a framework is developed in which robots become functional tools that assist the human, rather than replacing the human operator [17]. This effort is driven by the belief that machine learning, especially when implemented in complex integrated systems, needs to be evaluated on realistic tasks with a human in the loop [18].

- the lack of empirical and mathematical representational models for a common evaluation metric that assesses the performance of the human-robot team, where one or more robotic agents can be involved in the task completion process, and where such involvement can be sequential or parallel, and with varying levels of task dependency.

1.2 Objectives and Contributions

Several objectives and contributions are addressed in this thesis, the main ones being:

- proposing a further step toward generalizing a common performance metric framework for assessing human-robot team performance. Previously, a wide range of performance metrics were proposed in the literature, however, such metrics lacked generalizability due to a bias toward application-specific measures. In addition, most research focused on assessing the performance of either robots or humans, and ignored the capabilities and limitations of the other team member. Therefore, this work provides a foundation upon which we can assess how well the human and the robot perform as a team. Two crucial factors are therefore addressed: the human trust in automation, and the human reliability.

- proposing a two-level trust evaluation model which evaluates and assesses the human trust in automation. Modelling this key factor is essential in determining the nature of the relationship between the human user and the robot. Our proposed model combines the advantages of fuzzy logic and finite state machines to best model this phenomenon. The model significantly reduces the system complexity and the size of the knowledge base by grouping perceptions into first- and second-order.

- proposing a time-based human reliability assessment model that uses a finite fuzzy state machine to estimate the human reliability state. First-order Sugeno-like consequents are used for defuzzifying the active human reliability states. First-order consequents were used because human reliability degrades naturally with time even when the task complexity is simple and imposes only light physical and cognitive loads on the human operator.

- proposing intuitively-derived mathematical models that generalize our proposed common performance metric to accommodate for multi-robot systems - in which multiple robots cooperate, with the guidance of a human operator, toward completing some tasks. Such cooperation can occur in sequence or in parallel, and at different levels of dependency. All scenarios are considered, and an intuitive extension of our proposed common metric is presented, and supported with experiments and simulations.

- proposing an application robotic platform, in which robots and human users cooperate toward achievement of collective tasks, with varying levels of complexity and success. User feedback is noted at all times, to best model the human expert's knowledge and tune rules where needed.

Detailed descriptions of each of the presented objectives and contributions are addressed throughout this thesis.

1.3 Thesis Organization

The remainder of this thesis is organized as follows: chapter 2 reviews recent advances in the area of robotics and human-robot interaction, along with the state of the art of human-robot performance metrics presented in the literature, with some more focus on some important ones that we will make use of throughout this work. Chapter 3 provides a detailed description of the generic proposed performance metric framework, and its different building blocks. Following that, some important concepts and definitions that will be essential for subsequent sections are presented. Then, we present our new proposed "human trust in automation model", followed with our proposed "human reliability"model. This chapter also discusses the extension of the proposed metric to accommodate for multi-robot systems. Sequential, parallel, dependent, and independent types of interaction between the different robots are addressed. Chapter 4 discusses some preliminary simulation results of the proposed framework. Chapter 5 describes the experimental setup along with the detailed implementation of the application robotic platform. Chapter 6 presents further experimental results that support the validity and the correctness of the proposed knowledge base. Finally, chapter 7 concludes this thesis by summarizing the contributions of this work and suggesting ideas for future research work.

Chapter 2

Background and Literature Review

The design of performance metrics for task-oriented human-robot interaction systems is basically concerned with systems whose performance is to be evaluated and assessed, and then compared with other systems from similar or different application domains. Typically, the main agents of such systems are the robots and the human users, which cooperate toward accomplishing some collaborative tasks as a team. The importance of developing performance metrics emerges from the rapid growth in the fields of robotics and human-robot interaction (HRI) systems. Therefore, in the next sections, not only will the state of the art performance metrics discussed in the literature be outlined, but we will also present the state of the art advances in the areas of robotics and HRI systems, and hence emphasizing the importance of designing a generalized set of metrics that is able to assess such performance.

2.1 Advances in Robotics and Human-Robot Interaction Systems

Research on robotics has experienced an exponential growth in terms of theoretical foundations [19], [20], [21], and design of various higher-order abilities for robotic systems [22], [23], [24], [25], [26], [27], [28], especially in recent years. Unlike traditional robots, where the user gives the instruction and waits for the completion of the task, most of today's robotic systems require a high level of interaction between the robot and the human, especially for applications related to health care and services in general. In some robotics design methodology, systems are formed by a small group of robots, each with specific and limited functionality and working collectively. These robots will each cooperate with the user and with each other to provide a powerful and robust system. The robot and the user must then be able to communicate and cooperate in a straightforward manner. The robotic system, in an optimal way, should combine the intelligence of the operator as well as the artificial intelligence it was given in order to optimize its actions [29]. This requires effective communication between the robot and the operator through an interface. The interface must be capable of indicating the intentions or internal state of the robot to the user and enabling the user to send commands to the robot using natural ways (e.g. intuitive verbal commands and various somatic gestures) [30], [31]. In case of collaborative robots, robot-to-robot communication is required, which is more challenging to define. Kawamura [32] proposes to do it in such a way that the user is able to monitor the operation of the system and intervene at any point.

Such natural multimodal human-robot interaction has been growing fast for the last few decades. Interaction between humans and robots is crucial. This interaction is facilitated when proper visual and tactile sensing are combined, and human-robot communication is based on natural language, which is of central importance especially in the field of human friendly robots and humanoids. Thus, robots are required to be able to perceive, understand, and learn from all the modalities used by humans during face to face interaction, and act accordingly [33]. In fact, speech comes to be one of the most prominent tools used by humans. Pointing gestures, facial expressions, head poses, gaze, eye contact, and/or body language are all of great importance, and should be included in such modalities [34].

Speech-based interaction is prominent in man-machine interaction, but alone it is not sufficient. Vision-based interaction to recognize gestures, and vision capabilities for real world information about the objects mentioned in the speech, come to be of similar importance and tend to complement the speech information [35], [36], [37]. Throughout the way, natural human-robot communication, combined with appropriate vision systems, can help the robot to navigate and learn from its environment, including humans. Take the scenario where we choose to represent and classify objects according to their attributes, such as colour and shape. The vision system seeks to locate regions in the surroundings that share similar or identical attributes to those of the target object. Assuming that a *tennis ball* is represented as a yellow-coloured round object, when the robot is asked to get the tennis ball, colour segmentation and shape detection processes will be initiated by the robot. If such a yellow round object is detected, the robot prompts the user for a speech confirmation about the successfulness of the mission. Otherwise, the robot can communicate to the user about its current visual results through speech, along with any possible previous and/or current difficulties faced during the search process, and wait to see if the user's reply may provide further details that help to recognize the object [38]. Another possible scenario could emerge when the robot spots two possible tentative candidates, thus, prompting the user for further information about the actual location of the object.

Visual perception of some attributes related to the user himself, such as his location, posture, and focus of attention, can be at the same level of importance, and used to solve ambiguity in the user's speech, and to understand the human intention within a dialog situation. One example could be when the user commands the robot to "take this to the kitchen". In this situation, the robot needs to understand what "this"means or points to. Therefore, the 3D positions of the user's head and hands could be extracted, along with their head orientation and the directions of the pointing gestures in order to precisely determine the user's line of sight and locate the object "this". Such a process is highly motivated by the fact that humans tend to look at the pointing target while performing the gesture, and speaking about it [39].

Human-robot interaction is becoming a key part of today's technical systems. It is the interdisciplinary study of interaction dynamics between humans and robots, with contributions from the fields of human-computer interaction, robotics, artificial intelligence, and natural language understanding, and the principles of psychology, communication, and ethics. It addresses how humans interact with robots, and how best to design and implement robotic systems that are capable of accomplishing collaborative tasks in human environments safely and effectively [4]. This field of human-robot interaction has been recently receiving considerable attention in the academic community due to the rapid advances in the field of robotics. This has made it possible to use robots in a growing number of roles, not only in industrial and factory automation, but also in search and rescue, social and home services, entertainment, rehabilitation and medical care, military, and exploration, where robots are becoming more involved in increasingly more complex and less structured tasks and activities, that require indispensable interaction with people to complete the required tasks.

Much work is being directed toward the goal of designing human-friendly robots that can be safely operated and easily instructed. Vision, touch, and natural language are major components in realizing such human-friendly robots, and each one of them is being studied independently as they represent research areas in themselves [40]. Jijo-2, for example, a talking mobile robot, is able to build a probabilistic map of its office environment by acquiring missing location information through conversational dialogues with people using speech, and vision for navigation [41].

Recent advances in robotics and human-robot interaction have made it possible to design mobile robotic systems to aid rescue workers in urban search and rescue (USAR) operations [5], [6], [42], [43], [44], in which the fundamental purpose is to find and rescue victims (when a natural or anthropogenic disaster strikes) as efficiently and safely as possible, ensuring that human rescuers' lives are not subjected to great risk situations. Such conditions usually dictate that not much *a priori* information can be precisely given to robots about the environments, which makes it extremely difficult for robots to autonomously navigate the scenes, identify landmarks, and find victims. Therefore, a human operator in the loop to help guide a robot remotely is usually the case in most current applications of mobile robots in USAR operations. For example, small size robots that may carry cameras, hazardous material detectors, and/or medical payloads, can enter voids which are too small, dangerous, or deep for a human rescuer, and begin navigating larger voids that rescue workers are not able to enter until a fire has been extinguished or the structure has been reinforced, or search ahead of rescue teams, reporting conditions that may be hazardous [45].

Another major application area of human-robot interaction systems can be found in service and assistive robotics, which includes a very wide spectrum of application domains, such as office/house assistants [7], rehabilitation robots [8], [46], [47], wheelchair robots and mobility aides [48], [49], manipulator arms for physically disabled people [50], [51], companion robots [52], and educational robots [53]. In rehabilitation robotics, as in post-operative cardiac surgery recovery [54], or a post-stroke rehabilitation [8], researchers focus on enabling robots to fulfill the role of a coach, nurse, or companion, providing personalized encouragement and guidance, motivating and monitoring the user during the process of rehabilitation therapy, and guiding users to perform physical therapy exercises. A variety of assistive robotic systems that provide support for those who have age-related challenges have been also studied [55], [56], [57]. Robots in this area focus on assisting elderly people achieving physical tasks that they may not be able to do, including getting in and out of bed, adjusting a bed for maximum comfort [58], and/or doing chores around the house. A robot like Domo [59] could help elderly or wheelchair-bound people with simple household tasks like putting away dishes. Other HRI systems have been used as companion robots in the public areas of nursing homes. Huggable [60], a robot equipped with several sensors to detect different types of touch, NurseBot [56], a robot that is used to guide users around a nursing home, Paro [61], [62], an actuated stuffed seal that behaves in response to touch and sound, all attempt to provide the benefits of pet-assisted therapy in nursing homes that cannot support pets. Other HRI systems are being studied to assist diagnosing and providing therapy of children with autism spectrum disorders (ASD) [63], [64], [65], [66], [67], where robots prove to be a more comfortable social partner for children with ASD than people, providing a possible therapeutic role to improve social interactions and encourage social behaviors, such as talking, dancing, singing, and playing, with other children or parents.

Other applications of HRI interaction systems emerge in many examples of entertainment robotics, including the use of robots as dance partners [68], [69]. The police and military have their own share of HRI, where applications include gathering information to support a dangerous task, such as using remote vehicles in front line areas - to minimize risk exposure to soldiers - or bomb disposal [9]. Robots also have long been a part of space exploration. For example, the remarkable success in space robotics includes the exploration of the surface of the moon followed by more recent NASA success in exploring the surface of Mars [70].

All these facts make it clear that human-robot systems are everywhere nowadays, conducting tasks that vary from entertaining to cooperating with and serving humans. This emphasizes the real importance of designing a common performance metric to judge the effectiveness of the tasks being completed by the robot and the human as a team; otherwise, most of these systems will run the risk of being underappreciated or even ignored.

2.2 Literature Review on Existing Performance Metrics

Many systems have been implemented toward achieving effective human-machine collaboration or interaction, but run the risk of being ignored, if appropriate performance metrics that allow comparing the actual practical results with reference to standard accepted procedures are not in place. Therefore, presenting a generalized set of metrics that assesses the performance of the human-robot system becomes crucial. Mission or task effectiveness (TE) is one of the most popular and best-known metrics used to evaluate the performance of human-robot teams. It is a measure of how well a task is actually performed. TE can be (1) time-based that measures the speed of performance or the time needed for a successful completion of a specific task, (2) error-based which attempts to estimate or measure the number of mistakes or damage occurred while completing a task, or (3) coverage-based that measures how much of some larger goal is achieved [13]. For example, in driving or navigation scenarios, task effectiveness might be a measure of the time required to drive from point A to point B. In search tasks, TE could measure the time needed to find all targets, and/or the number of targets found in a given amount of time. In an assault, TE might measure the number of targets destroyed or estimate the losses taken.

Although successful deployment of TE performance metrics can be found in many scenarios and applications, such measures face several important problems:

- measures of current task effectiveness can be very misleading. A robot might currently appear to be effective, but on the overall goal, it might be making negative progress. For example, a robot might be getting closer to the target very rapidly and yet be wandering into a cul-de-sac or a dead end from which it will need to back out [71].
- measures of current task effectiveness fail to provide insights into the process that leads to the final mission-related output; hence, focusing on just the mission effectiveness makes it difficult to extract information to detect design flaws and to design systems that can consistently support successful mission completion [13].
- measures of current task effectiveness are not sufficient to understand team performance issues and to identify design improvements, and hence additional metrics are required [72].
- measures of current task effectiveness can be highly task-specific: for example, task completion time fits many robotic applications, such as retrieving an object with a robotic manipulator, or navigating from point A to point B. However, it may not suit other applications, such as a range of motion exercise in the rehabilitation of an upper limb [16].

Situation awareness also finds itself as another emerging metric used in the literature to assess systems performance. Endsley [15] defines situation awareness as "the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future". Following this definition, we can notice that Endsley defined three levels for situation awareness [73]:

- perception: basic perception of important information. This ability allows the robot to receive multi-sensory input from the external environment. Attention is the main mechanism required in this process, which adopts only that portion of input stimuli which is relevant and useful with respect to the current target and context of the robot. Successful operation of the robotic system is highly influenced by the efficiency of the attention mechanism deployed, as it avoids the system being flooded with an enormous amount of unnecessary information. Furthermore, perception is connected to all other components of the system, and is therefore capable of sensing the internal state of robot, a central requirement of realizing self-awareness capability in intelligent robots.

- comprehension: correct interpretation and integration of perceptions as well as relevance assessment. This level is highly related to knowledge representation, where a well-designed knowledge representation scheme facilitates ease of information storage and retrieval and performing inference to obtain new information from learned or embedded data. The exact form of knowledge representation, however, is still a matter of debate.

- projection: the ability to reason and predict future situations based on current perceptions and background knowledge. This component is comprised of a set of processes functioning simultaneously and potentially interacting with each other. It involves high-level reasoning where the robot computes how to perform a given task most efficiently based on its abilities, learned knowledge, embodiment, and situatedness.

Situation awareness, thus, requires an intelligent robotics system to: (1) have a properly designed knowledge base that can be queried for known and unknown information, (2) be aware of its sensors and the kind of information they represent, (3) deploy an effective attention mechanism that identifies, classifies, and selects important perceived information based on a current set of goals, which avoids the system being flooded with an enormous amount of unnecessary information, (4) have knowledge on information dynamics that allows for reasoning and prediction of future states based on the current situation and previous knowledge, and finally (5) have and gather (learn) information beyond their sensory capabilities by cooperating with other agents for external information acquisition [73]. That said, task-specific metrics are to be designed in order to assess the performance of each the mentioned objectives.

Aside from situation awareness, other studies chose to focus on task-oriented mobile robots. Steinfeld et al. [10] present a set of special metrics that assess five task categories: navigational, perceptual, management, manipulatory, and social. By doing so, they believe that: (1) their metrics are broadly applicable to a wide range of applications, and (2) they can assess the impact of different levels/types of HRI on performance. Examples of some of the presented metrics are: percentage of navigation tasks successfully completed, coverage of area, deviation from planned route, obstacles that were successfully avoided, obstacles that were not avoided, but could be overcome, absolute and relative judgments of distance, size, or length, time or effort to confirm identification, detection accuracy for targets within sensor range, degree of mental computation, intervention response time, level of trust, engagement, and compliance [10].

Many other metrics have been presented in the literature, however most of them are domain- and application-specific, and lack the generalization aspect, which made them not comparable due to a bias toward application-specific measures. Examples of such metrics are those that address assessing the performance of assistive robotics systems, such as systems that help people with autism spectrum disorder, provide elders with care and assistance, and other assistive systems such as intelligent wheelchairs, assistive robotic arms, and external limb prostheses [16]. Hence, designing more generalized performance metrics that can translate well between applications becomes more important.

The idea of developing a common toolkit of performance metrics and identifying measures that can be used to compare different research results has also been discussed by other researchers. Olsen and Goodrich discuss six interrelated performance metrics that can lead the design of human-robot interaction systems. They claim that these metrics are somewhat generic, and together they provide a framework for assessing the interaction design [71], [74]. Such metrics are: task effectiveness (TE), neglect tolerance (NT), robot attention demand (RAD), free time (FT), fan-out (FO) and interaction effort (IE).

- Task Effectiveness (TE): discussed earlier, this metric is a measure of how well a task is being accomplished by the human-robot team. Several measures can be used for this purpose; time-based metrics measure the speed of performance, error-based metrics measure the size of error and damage, and coverage-based metrics measure how much of the overall goal has been accomplished.

- Neglect Tolerance (NT): this metric is a measure of how the robot's current task effectiveness drops over time when being neglected by the user. This metric is an important measure of the robot autonomy with respect to some task. Olsen and Goodrich present two ways to measure neglect tolerance. In the first scenario, neglect tolerance is a measure of the average neglect time. The robot is placed in a random location in the problem world, and given a specific task to complete; then the amount of time in which the robot performs effectively - meaning that the robot is making progress toward the completion of the final goal before its performance drops below a predefined effectiveness threshold, as in stopping and not knowing what to do next - is measured. However, such a measure might be wrong; the robot might seem to be making progress toward the final goal, when it is just wandering into a cul-de-sac. Therefore, neglect tolerance is not as simple, as it should also involve the user interface and the global problem space. Frequently the users will detect global problems, and will intervene before the robot itself detects the problem. Therefore, an alternative and more efficient way to measure the neglect tolerance is to rely on the human's estimate of the current robot progress toward completion of the task. In this scenario, NT is a measure of the time elapsed between the human instruction to the robot, and either a drop of the robot's performance below the effectiveness threshold, or the next human instruction in case the user detects a problem. This leads to a more logical and accurate measurement of such neglect tolerance metric.

- Robot Attention Demand (RAD): this metric is a measure of how much time or fraction of the total task time the user must spend toward interacting with the robot. Interaction effort (IE) is a key component in determining this metric, as will be addressed shortly. RAD is defined as a relationship between NT and IE as shown in equation 2.1, where the numerator is the amount of effort that the user must spend interacting with the robot, and the denominator is the total amount of effective time of the robot.

$$RAD = \frac{IE}{IE + NT} \tag{2.1}$$

- Free Time (FT): this metric is an extension of the RAD notation. It represents the amount of free time in which the user is not interacting with the robot, and hence, he/she can spend this free time doing something else, such as interacting with another robot toward achieving a different task. FT can be obtained by subtracting the RAD from the total task time, as shown in equation 2.2.

$$FT = 1.0 - RAD \tag{2.2}$$

- Fan-out (FO): this metric is a measure of how many robots a user can operate and interact with simultaneously and effectively. This metric is defined as the total task time divided by the time spent interacting with one robot (RAD), as shown in equation 2.3. The equation shows that NT and FO are proportional, and hence when the NT increases, FO also increases; however, this is not the whole story, as it is also a fact that when FO increases, IE also increases.

$$FO = \frac{1.0}{RAD} = \frac{IE + NT}{IE} \tag{2.3}$$

  An alternative way to measure FO is by measuring the average number of robots operating above the effectiveness threshold while interacting with the human. In this scenario, the user is given a number of robots, and while the task is being progressively completed, the number of robots operating effectively is counted, and the FO is reported as the average count of such robots.

- Interaction Effort (IE): in most cases, IE is a measure of the time required to interact with a robot, and reducing this factor is a key problem in human-robot interaction systems. However, defining this metric is not as easy, because in most scenarios, interaction effort is rather cognitive than physical. Hence, in order to solve this problem, interaction effort and not only interaction time should be measured: some comparative tools for measuring efforts and progress should be used. An alternative approach is to experimentally measure NT and FO, and then approximate the IE using equation 2.3, which leads us into equation 2.4.

$$IE = \frac{NT}{FO - 1} \tag{2.4}$$

Although the presented set of metrics proposed by Olsen and Goodrich are somewhat generic, they lack focus on two essential factors that largely impact the human-robot interaction process. Those factors are: human trust in automation and human reliability. More emphasis on those factors will be presented in subsequent sections. Therefore, a metric framework that can be generalized should also involve the human trust and the human cognitive limitations in the human-robot interaction performance assessment loop [72]. In this work, we intend to present another further step toward presenting a generalized common metric, that attempts to model both the performance of the human and the robot as a team [75], [76], [77], [78]. This model is motivated by Olsen and Goodrich's work [71], [74] toward designing a generic metric for assessing human-robot team performance, while addressing all the shortcomings and drawbacks of the current proposed set of metrics. Toward this goal, we model two important human factors that are essential to the human-robot performance assessment. A two-level fuzzy temporal model to estimate the human trust in automation level is presented, while another fuzzy temporal model is also proposed to estimate the human reliability during human-robot interaction time. Details on architecture and implementation will follow in subsequent sections.

Another important issue that Olsen and Goodrich, among many other researchers, ignored, is when the robotic system is composed of multiple robotic agents instead of just one. In fact, a one-robot system should be treated differently than a two-robot system, or a three-robot system. For instance, addressing the RAD as described by Olsen, would a two-robot system have twice the RAD compared to a one-robot system? If so, can we still use it as an indicator to how well the overall system is performing? The answer is short and clear, of course not, because if we just ignore the fact that these two robots are a part of a bigger system, working toward achieving a bigger goal, then we are restricting this metric from being used as a metric to assess the performance of that team. What we need is a metric to assess the performance of the whole human-robot team, and therefore, for example, a three-robot system successfully completing its task should have a performance measure or index that is close in value to that of a one-robot system that is successfully completing its task as well, although the RAD in the first case might be nearly three times its value in the second case. Therefore, the performance metric should evaluate the overall performance of the whole system, and be a good indicator of how well the team is performing (the human user and the robotic system). This means that a special consideration to multi-robot systems should be also addressed. Several cases emerge from this fact, as these robots can have sequential or parallel ways of executing their tasks, and with different levels of dependency; hence, each case should be individually considered. Therefore, in this work, we also extend and generalize our proposed generic metric framework to accommodate each of the previously mentioned cases.

2.3 Chapter Summary

Research on robotics has experienced an exponential growth in terms of theoretical foundations, and design of various capabilities. Unlike traditional robots, where the user gives the instruction and waits for the completion of the task, modern systems require a high level of interaction between the robot and the human, where robots are becoming more involved in increasingly more complex and less structured tasks and activities that require indispensable interaction with people to complete.

Therefore, presenting a generalized set of metrics that assess the performance of the human-robot system becomes crucial.

The idea of developing a common toolkit of performance metrics is discussed by Olsen and Goodrich, who present six interrelated performance metrics that can lead the design of human-robot interaction systems [71], [74]. Such metrics are: task effectiveness (TE), neglect tolerance (NT), robot attention demand (RAD), free time (FT), fan-out (FO) and interaction effort (IE). Such metrics, however, lack focus on two essential factors that largely impact the human-robot interaction process: human trust in automation and human reliability. Therefore, proper modelling of such factors is crucial. A special consideration to multi-robot systems should be also addressed. Sequential and parallel robot cooperation schemes with varying levels of task dependency shall be considered.

Chapter 3

Proposed Performance Metric

This work builds on top of the work presented by Olsen, and Goodrich [71], [74], and the most recent state-of-the-art qualitative performance measures in the area of human-robot interaction. It proposes a common generic metric framework to assess the performance of the human-robot team. In this chapter, we propose a common metric, and identify and model the key factors that are crucial to define it. Doing so, the focus is first drawn to one-robot systems, in which a human user is collaborating with only *one* robot toward achieving some well-defined tasks. This metric is then extended and generalized in the subsequent sections to accommodate for the scenarios where multiple robots can be a part of one bigger system, in which they collaboratively achieve tasks toward the fulfillment of the final goal. Several cases are considered in this scenario, where sequential and parallel execution of tasks can take place, with varying levels of dependency. Such scenarios are addressed, and appropriate extension models for our presented metric framework are proposed for each of those cases.

As mentioned before, one big constraint on the fan-out metric is caused by the limitations of human cognition and memory reliability when interacting with multiple robots; the user must probably remember not only the current robot situation, and its corresponding information, but also the interface modes, robot capabilities and limitations, as well as some history of interaction with that robot, the goal being worked toward, and/or the fraction of work that has been completed. Therefore, modelling this human efficiency (or human reliability as we will call it) and involving it as a key factor in determining a more accurate measurement of FO is critical.

Another key factor will also be discussed. In this chapter, we describe the trust in automation as a key factor for social robots and in assessing team performance. This factor is essential in determining the nature of the relationship between the human user and the robot, thus properly modelling it is also essential. In this work, we focus on both RAD and FO, as we believe that both metrics are missing important key factors, and should be augmented. Human trust in automation and human reliability models shall be presented. Fuzzy temporal models are proposed to estimate the contribution of both key factors in determining the final value of the generic metric.

3.1 Overall Proposed Metric Framework

In this section, we present our alternative definitions of both the FO and the RAD metrics by further including the human reliability (HR), and the human trust in automation (Tr) factors. Modelling both factors will take place in subsequent sections in this chapter.

3.1.1 Fan-out and Human Reliability

Fan-out, a metric originally defined by Olsen and Goodrich, is a measure of how many robots with similar capabilities a human can simultaneously and effectively control and/or interact with. This definition is directly related to the robots' control demands, management difficulties during use, and the total cost-benefit ratio of the robot system [10]. The fan-out metric is defined in terms of RAD as shown in equation 2.3.

However, there are several cognitive and physical constraints and limitations that make it hard to measure this fan-out limit. The first limitation is caused by the task saturation. In this scenario, the task space becomes either too crowded, and sending more robots will not help increasing the performance, or saturated, meaning that the task is too simple and does not require many robots to achieve, and hence sending more robots to complete a simple task will not lead to any further increase in the performance. For example, a task with only two simple targets that requires only two robots to complete will not be performed better when we send ten robots to complete this task. Add to this the complexity of having a crowded task space, which might also negatively affect the task completion. Hence, measuring this FO factor experimentally becomes a difficult task. The second important and critical limitation is based on the human cognitive and physical limitations. When the task space becomes crowded, the user has probably to remember the current state information of all the robots, some interaction history, and/or the capabilities and limitations of each robot. Hence, human reliability becomes a serious issue, which clearly varies significantly with task complexity, number of subtasks being completed, mental workload, external and internal burden (such as stressing work environment - e.g. hot temperature), the psychological and sociological situation of the human operator, among many other factors. Thus modelling this HR factor becomes a necessity. Therefore, an alternative definition of the FO metric is proposed as shown in equation 3.1, where HR, a value between 0 and 1, represents the human reliability, and the ability of the human operator to manage the increased complexity of the system as the number of simultaneously active robots increases.

$$FO = \frac{1.0}{RAD} \times HR \tag{3.1}$$

Human reliability is assumed to be one of the major issues and limitations in measuring the attributed human role in complex human-robot interaction systems. The reasons behind these limitations are quite obvious, ranging from the many interacting variables such as workload (which is also dependent on many other variables, such as the environment and task complexity), skill, which also depends on the level of training that the human operator has been subject to, as well as the levels of expertise, and many other interrelated factors and variables [79], [80]. Because of these limitations, human reliability is often measured using experimental simulation settings [81]. In this work, as presented in subsequent sections, we present our proposed fuzzy temporal model for estimating the human reliability, in an attempt to include this critical factor in the human-robot team performance assessment technique. Several important components are to be taken into consideration when talking about human reliability analysis [81], such as:

- workload associated with the different tasks, and the mental workload required for managing them. [82]
- task difficulty, which also includes attributes on the physical, physiological, and sociological situation of the human operator such as stress, fatigue, boredom, family and social problems, and so on [83].
- human skill, which is based on the levels of training, experience, and education that the user has [84].
- intrinsic error rate, that depends on the constraints and limitations of human motor sensory, psychological, and cognitive faculties [85].

- time factors which basically represents the effect of continuous work on performance [81].
- external burden, such as a difficult/harsh working environment, or uncomfortable weather conditions [85].

3.1.2 Robot Attention Demand and Human Trust

Olsen [71] defined robot attention demand (RAD) as a measure of the fraction of total task time that a user must spend interacting with a robot. RAD was defined as a relationship between neglect tolerance (NT) and interaction effort (IE) as presented in equation 2.1. Olsen presents a more accurate way to approximate the value of the neglect tolerance by measuring the time between the human instruction and either a drop in robot performance below the effectiveness threshold, or the intervention of the human with another instruction. However, Olsen also states that in this scenario, this metric is no more independent from the user, thus the operator's trust in the robot's autonomous abilities becomes a critical issue that can highly influence the interaction process.

In this work, we attempt to determine the true time that an operator has to dedicate to the robot. Therefore, we present an alternative definition of the RAD as a function of both direct interaction time (DIT) and indirect interaction time (IIT). The IIT is a direct consequence of trust, and can represent the time being spent when the robot is being neglected completing some tasks, but still with much of the user's attention drawn to it as a result of the operator's distrust in the machine. This relationship is shown in equation 3.2, where NT represents the neglect tolerance, and Tr is the human operator's trust in the robot. A two-level fuzzy temporal model to estimate the trust value will be presented in subsequent sections.

$$RAD = DIT + IIT = DIT + NT \times (1 - Tr) \tag{3.2}$$

Trust is an old phenomenon that has been extensively studied in the literature of sociology, social psychology and philosophy, because of its importance in societies and interpersonal relationships. The origins of such work can be found in [86], [87], [88], [89]. Golembiewski et al. [90] states that "perhaps there is no single variable which so thoroughly influences interpersonal and group behaviour as does trust"[91]. Trust in automation is also not a new concept. It has been also extensively discussed in the literature by many researchers, especially in the fields of human-machine interaction [92], [93], [94], [95]. Trust in human-robot teams is a key factor in determining the success of such a team, as much as it is significant for determining the success of a human-human team [96].

Other research studied trust in process-control systems [97], [92], [98], [99]. For example, Lee et al. [100] examines the relationship between trust in automatic controllers and the user's self-confidence in manually operating a simulated semi-automatic pasteurization plant. Muir et al. [98] present some experimental studies that examine the relationship between trust and human intervention in a process control simulation. The results showed that operators' subjective evaluation of trust in the machines was based mainly on their perception of the robot's competence. Trust significantly dropped when the robot showed slight signs of incompetence, even when it did not affect the overall objective. There was a high correlation between operators' trust and use of the automation, where operators chose to rely on manual operation when the trust in automation was low. The results also showed an inverse relationship between trust and monitoring of the automation. Operators tend to actually neglect the robot when they are not interacting with it, when their trust in its automation is high. The study also shows that trust slightly changes with experience, and distrust in one function spreads to reduce trust in another functions [101]. Similar findings were also reported by Zuboff [102] and Sheridan [97]. Since then, various researchers have tried to understand the role trust plays in system performance for a wide range of complex automated systems, such as air traffic control [103] and antiaircraft warfare [99].

Various other studies studied trust alongside the self-confidence of the human operator, and found that they correlate [100]. As the user's self-confidence goes down, trust in automation goes up, thus resulting in an increased use of the automation; when self-confidence goes up, trust in automation goes down, resulting in a decreased use of automation [96]. Other researchers also tried to formulate a model for trust between humans and machines. Lee [100] fitted a time series model and found a relationships for trust in a feedstock pump as shown in equation 3.3, where T refers to trust, P to productivity, F to fault size, C to some weight coefficient, and v to residual error. Moray, working with a simulated air-conditioning plant called SCARLETT, also found similar time series equations [104].

$$T_n = C_1 T_{n-1} + C_2 P_n + C_3 P_{n-1} + C_4 F_n + C_4 F_{n-1} + v \tag{3.3}$$

These studies make it conclusive that trust is a key factor in determining the type of interaction between humans and machines, and therefore, has great implications for their collaborative performance as a team. Several factors lead to trust development between the operator and the robot; some main ones as reported by Madsen [105] are:

- reliability, and the ability of the system to maintain consistent functioning.
- robustness and ability to perform effectively under a variety of circumstances and external environmental settings.
- familiarity, in terms of employment of procedures that are friendly and familiar to the operator.
- understandability and predictability, meaning that the human operator is able to formulate a mental model that is able to predict future system reasoning and behavior.
- clarity of intention, meaning that the system displays and explicitly explains to the operator its current state and what it intends to do in its next step. This step is pretty much an explicit explication of the inferred decisions to the user before taking any further actions.
- technical competence, meaning that the system is able to accurately and correctly perform the tasks based on the new input information and current knowledge base.
- integrity, which is the ability of the system to recover from technical failures with minimal loss and damage.
- personal attachment to the system, as an example when the user finds the system suitable to their personal taste, and therefore develop a strong preference for using it.
- faith, meaning that the user has faith in the system's future ability to perform effectively even in unknown or never previously encountered situations. This factor is highly affected by the reputation of the system itself, which makes users develop a stronger faith in its ability.

Therefore, and based on the above criteria and factors, we can clearly see that the trust varies with time depending on the reliability, robustness, technical competence, integrity, and learning capability of the robot. It also depends on the personal attachment and the faith that each human might have for the designated robot. It can thus be conclusively stated that the more errors and mistakes the robot makes, the less trusted it will be by the operator, and vice versa. The more productivity and utility the robot produces and achieves, the more it will be trusted.

Add to this the fact that the more the robot is self-aware of its abilities and limitations, human-aware of the human availability and reliability, context-aware of the task being completed in a certain environment, the more trust it will gain from the operator.

3.2 Proposed Trust Fuzzy Temporal Model

The previous discussions on trust and its evolution, and all the attempts to quantify this instrumental phenomenon in human-robot interaction teams, make it clear that coming up with a unique mathematical formula that governs the temporal behaviour of trust, which is even difficult to measure experimentally, is going to be far from the truth, unrealistic, and domain- and application-specific. This is because when we talk about trust, we talk about something intangible, insubstantial, vague, and not very clear to numerically define, something that is **fuzzy** in nature. From here came the motivation to take advantage of all the discussions, results, summaries, and even the mathematical attempts to describe and model this phenomenon, and get inspired about a new temporal fuzzy model that represents the main components that contribute to the evolution of the human trust in automation during interaction with the robot agent.

3.2.1 Definitions and Background Information

In the following, we present some background information on some tools and techniques that are instrumental to the rest of our proposed work.

Fuzzy Logic

Intelligent systems generally have a capacity to acquire and apply knowledge in an "intelligent"manner and have the capabilities of perception, reasoning, learning, and making inferences and decisions from incomplete information. The most well-known knowledge base systems presented in the literature are the fuzzy logic systems proposed by Zadeh [106]. The conventional binary logic is crisp, and allows for only two states: true, and false. This logic, however, cannot handle fuzzy descriptors, as in "fast", and "slow", which are qualitative, subjective, and descriptive, rather than quantitative, and may contain some overlapping degree of a neighbouring quantity, for example, some degree of slowness in the fuzzy quantity "fast"itself. Fuzzy logic allows doing a realistic extension of binary crisp logic, to quantitative, subjective, and approximate situations, which often exist in problems of intelligent machines. Therefore, instead of representing a system with a set of complex mathematical equations that can be very unrealistic, as in modelling trust for example, fuzzy systems use simple empirical rules to represent input and output relationships by applying the available human expert' knowledge.

Fuzzy logic is based on fuzzy sets in a similar manner to how crisp bivalent logic is based on crisp set theory. A fuzzy set is represented by a membership function. A particular element value in the range of definition of the fuzzy set will have a grade of membership, which gives the degree to which the particular element belongs to the set. Unlike an ordinary crisp set where each object or element either belongs to the set or does not belong to the set, partial membership in a fuzzy set is possible. In this manner, it is possible for an element to belong to the set at some degree, and simultaneously not belong to the set at a complementary degree. It is also possible for an element to belong to the set at some degree, and simultaneously belong to another set at some other degree. In other words, there is a softness associated

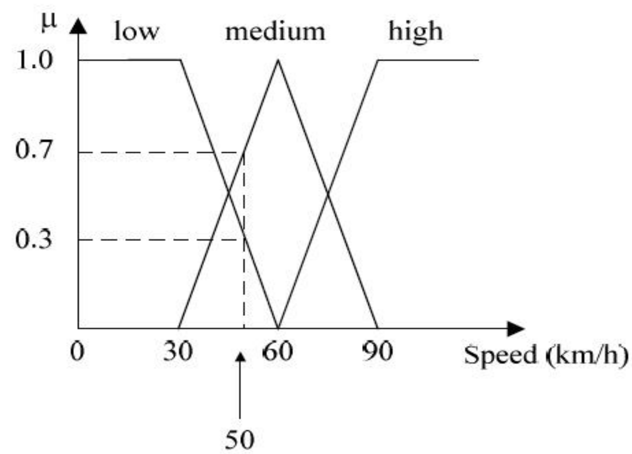


Figure 3.1: Sample Fuzzification Process

with the membership of an element in a fuzzy set. Figure 3.1 shows three fuzzy sets that correspond to the object speed. Speed value of 50 km/h belongs to the set "medium"with membership grade 0.7, and set "low"with membership grade 0.3.

A fuzzy set may be represented by a membership function. This function gives the grade or degree of membership within the set for any element in the universe of discourse, where the universe X is the set that contains every subset of interest in the context of a given class of problems, and whose elements are denoted by x. The membership function maps the elements of the universe x onto numerical values in the interval [0, 1] as shown in equation 3.4. This process is referred to as fuzzification.

$$\mu_A(x) :\rightarrow [0, 1] \tag{3.4}$$

Fuzzy Inference Systems

Fuzzy inference systems consist of four major components: fuzzifier, knowledge base, inference mechanism, and a defuzzifier [107]. Figure 3.2 illustrates the basic configuration of a fuzzy inference system. The first component, fuzzification,

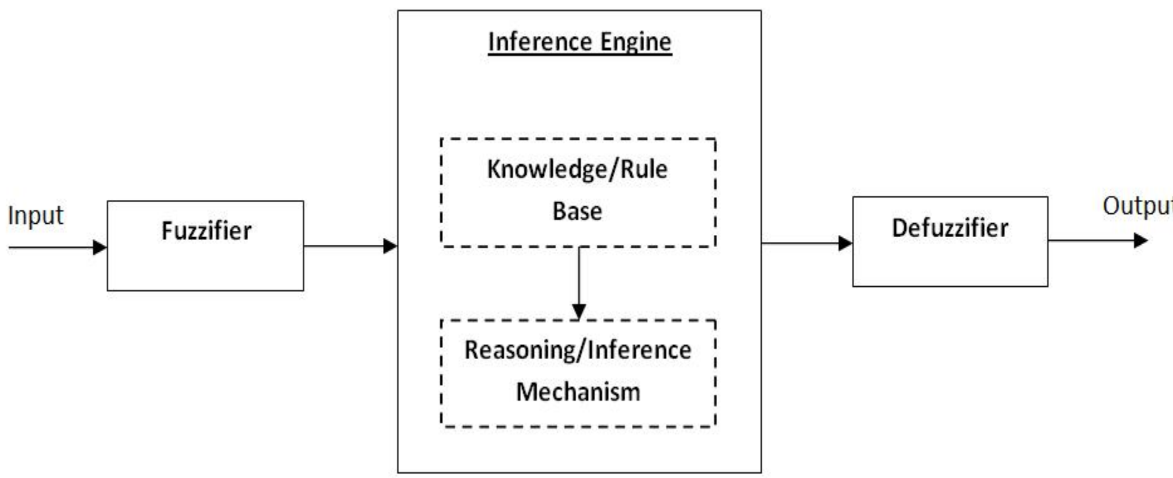


Figure 3.2: General Configuration of a Fuzzy Inference System

which was already described, represents the transformation or the mapping process of crisp input values to membership grades between 0 and 1 through the use of membership functions. The second component is the knowledge base, which is represented by a set of *if-then* rules of fuzzy descriptors. An example of such fuzzy rules would be:

"**IF** the speed is *slow*, **AND** the target is *far*, **THEN** *moderately* increase the power",
OR

"**IF** the speed is *slow*, **AND** the target is *near*, **THEN** *moderately* decrease the power",

where *slow*, *far*, and *moderate* are fuzzy descriptors that could be represented by some membership functions. The two most commonly used fuzzy operators in fuzzy rules are the AND operator (in the rule itself) and the OR operator (from one rule to another), which are generally referred to as the T-norm operator and the S-norm operator respectively. T-norm and S-norm operators are not unique. In fact, any operator that satisfies some corresponding boundary, monotonicity, commutativity and associativity conditions, can be classified as one of these norms. The most famous examples of the T-norm and the S-norm operators are the minimum and the maximum operators, respectively.

The third and the most important component in a fuzzy inference model is the fuzzy reasoning mechanism which attempts to model the human decision-making process. The well-known Generalized Modus Ponens (GMP) is applied to the knowledge base in order to achieve this fuzzy reasoning. GMP simply states that given the fact x is A' and the proposition or the premise "IF x is A THEN y is B", it can be concluded that y is B' [108]. In other words, given new inputs and combined with the knowledge-base, conclusions can be drawn in a process that involves two steps as shown in Figure 3.3:

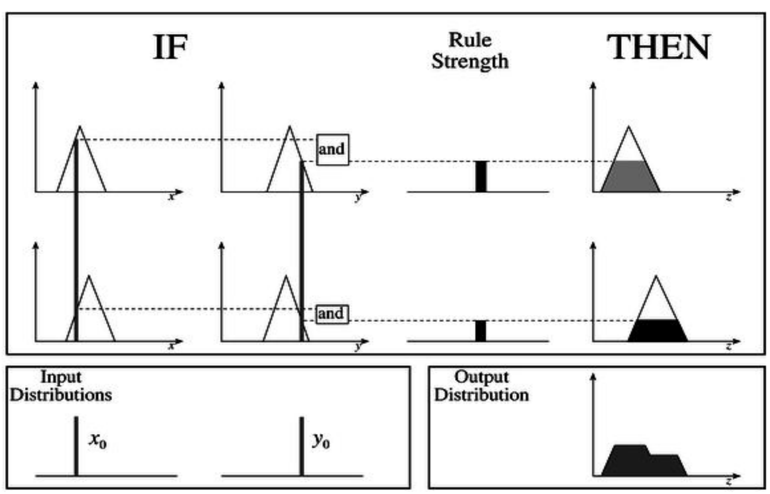


Figure 3.3: Inference: Rule Evaluation and Aggregation [1]

- rule evaluation: in the first step, decisions corresponding to individual rules in the knowledge base are computed. Evaluation of the antecedent is necessary. For an antecedent with a single part (as in "**If x is A**, THEN ..."), the evaluation result is represented by the fuzzified system input. For antecedents with more parts such as ("**IF** x_1 **is** A_1**, AND** x_2 **is** A_2, THEN ..."), fuzzy operations are applied to the membership values corresponding to the system inputs, where the most common operation is the *minimum* operation - hence providing a single numerical value, representing the level of truthfulness of the antecedent, which is also called the *firing strength* of the rule. This number, commonly termed as a weighting parameter, is utilized to adjust the membership function of the corresponding rule consequent (the conclusion part of the rule) for generation of the rule conclusion. A consequent membership function can either be truncated or scaled by the weighting parameter.

- rule output aggregation: the second step consists of combining all rule outputs to generate the final conclusion. In this step, all rule consequents are unified into a single fuzzy set through the fuzzy OR operator, which is commonly represented by the S-norm operation *maximum*.

Several inference models were presented in the literature, however the two most well-known and commonly used ones are Mamdani [109] and Sugeno [110] inference models. In the first approach, consequent membership functions are fuzzy sets. The inference process results in a fuzzy set that requires a process that is called defuzzification. However, unlike the Mamdani model, consequent membership functions in a Sugeno inference model are fuzzy singletons that have membership value of 1 at a single point in the universe of discourse, and a value of zero elsewhere. Consequents in the Sugeno model are usually a function of its rule antecedents. They are commonly represented by first-order or zero-order equations. In a zero-order Sugeno model, the consequent membership functions are basically constant values. An example of a fuzzy rule in Sugeno fuzzy inference model is the following:

IF target is near AND speed is fast, THEN deceleration is $-speed^2/\,2\,\times distance$

Sugeno fuzzy inference models do not require defuzzification of the final aggregated output. The crisp overall output, is computed using the weighted average method, according to the equation 3.5.

$$\hat{c} = \frac{\sum_{i=0}^{n} w_i c_i}{\sum_{i=0}^{n} w_i} \tag{3.5}$$

where w_i is the i^{th} rule firing strength, c_i is the rule consequent, and n is the number of fired rules.

The last component in a general fuzzy inference model is the defuzzifier, which maps output values from fuzzy sets back to crisp numerical values. In the Mamdani approach, a defuzzifier is required, while in the Sugeno fuzzy model, such a step is not required. Several defuzzification methods were proposed in the literature, and probably the most commonly used ones are the centroid method, and the mean of maxima [107].

Fuzzy State Automata

Fuzzy logic systems have been used in several control applications due to their interesting performance and ability to include human knowledge into a controller design; such systems, however, are feed-forward with no feedback, and therefore with no memory. This fact constraints the applicability of these systems in many applications, especially in the field of control, where most controllers must have memory. The fuzzy state automata presented in the literature [111], [2], [112] came to address these limitations by designing recurrent fuzzy systems with sequences of states and events.

Automata are mathematical models of computations. Several kinds of automata have been presented in the literature [113], [114], [115], [116], [117], and among them a finite automaton is the simplest and most known. A finite automaton can be seen as a finite control, in some state from a finite set of states, reading a sequence of symbols from a finite input alphabet. The machine is in **only one** state at a time, and this state is called the current state. In one move or transition that is triggered by some input symbol, a finite automaton in some state enters a new state, which is solely determined by the last state and the input symbol being scanned. In a finite automaton, the input alphabet consists of a finite number of discrete input symbols. These input symbols may be reasonably thought of as the input values that we are going to compute [118].

There are two different groups of finite state machines (FSM): acceptors and transducers [119]:

- acceptors, also called recognizers or sequence detectors, are finite state machines that generate a yes or no output to answer whether an input is accepted or rejected by the machine. If the final current state, after a whole input string is processed, is an accepting state, then the input is said to be accepted by the machine; otherwise it is rejected. An example of an accepting state is shown in Figure 3.4. The start state S_1 is an accepting state as it detects whether an input binary string contains an even number of $0's$ or not. Once an input string is processed, if the machine finishes at state S_1, the binary string is said to contain an even number of $0's$.

- transducers are finite state machines that produce an output and action based on a given input and state. Such FSM have many applications in the fields of systems control and computational linguistics. Two types of machine are highlighted: (1) Moore machines, in which the output depends only on the state, and (2) Mealy machines, in which the output depends on both the input and the state.

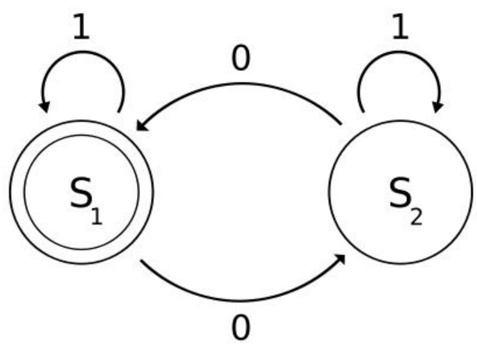


Figure 3.4: Finite State Machine

Fuzzy automata are generalizations of finite automata, first introduced by Santos [120] in the late 1960s. Afterwards, languages accepted by fuzzy automata were studied by Zadeh [121] and Thomason and Marinos [122] in the early 1970s, and the fundamentals of fuzzy automata were discussed by Gaines and Kohout [123] in 1976. In the early 1990s, the potential of fuzzy automata as design tools for modelling a variety of uncertain dynamic systems was exploited; various methods for synthesis, analysis, specification and implementation of fuzzy automata were proposed. For example, Giles, Omlin, and Thornber [124] presented a synthesis method for mapping fuzzy automata into recurrent neural networks.

In fuzzy automata, the knowledge about the system's next state is vague or uncertain; therefore, such systems can be seen as models of computing with values, but a certain vagueness or uncertainty is involved in the process of computation [118]. Fuzzy automata which combine the capabilities of finite automata and fuzzy logic are shown to be very useful for areas which are well-known to be previously addressed by discrete mathematics and probabilistic approaches [125], [126], [127]. Unlike finite state automata (FSA), finite fuzzy state automata (FFSA) transitions are not triggered by crisp inputs but by fuzzy variables; the same applies to state transitions as well. Therefore, the whole system does not necessarily have to be in only one well-defined state, but it may be in more than one state at the same time, with some corresponding membership grade or activation level associated to each active state.

A finite fuzzy state automaton $\widetilde{F}$ is a 6-tuple denoted as $\widetilde{F} = (Q, \Sigma, \delta, R, Z, \omega)$ [2], where:

- Q is a finite set of states, $Q = \{q_1, q_2, .., q_n\}$.
- Σ is a finite set of input symbols, $\Sigma = \{a_1, a_2, .., a_m\}$.
- $R \subseteq Q$, is the fuzzy set of start states of $\widetilde{F}$.
- Z is a finite set of output symbols, $Z = \{b_1, b_2, .., b_k\}$
- $\delta : Q \times \Sigma \times Q \rightarrow [0, 1]$, is the fuzzy transition function that is used to map the current state (also called predecessor) into a next state (also called successor) upon receiving an input symbol, producing a membership value in the fuzzy interval [0,1] to the next state.
- $\omega : Q \rightarrow Z$, is an output function which is used to map the fuzzy states to the output set.

Every state in a FFSM represents a temporal phase of the signal evolution in time, which is representative and descriptive of the application domain. We say the signal is in a specific state when its attributes fulfill some predefined constraints, which are fuzzy in nature; hence, the activation level of a state is a matter of degree, where more than one state can be active at the same time. Two crucial issues need to be addressed when completing a state membership assignment. The first one relates to how to produce a membership grade or a state activation level to a successor state upon the completion of a fuzzy transition. The second issue addresses the proper way to deal with the scenario when multiple transitions lead to the same next state, and hence the state is forced to take more than one membership value simultaneously.

State Membership Assignment

The fuzzy automaton has come a long way since its theory was first introduced. Originally, there was an approved approach for assigning membership value to a next state, where only the transition weights are taken into account, and therefore, the membership value of the current state is ignored [2], [112]. Thus, the next state activation level is considered to take the value of the weight of the transition that leads to that state. This technique is called the transition-based membership approach. This method however, suffers from some disadvantages that make it unsuitable for many applications. To better observe the negative consequence of such transition-based membership approach, we consider the example shown in Figure 3.5. The membership value of the current state q_1 of the fuzzy state automaton at time t is 0.01, and the weight of the transition from state q_1 to state q_2 upon input symbol a is 1.0. Using the transition-based membership method, and assuming that the input symbol upon time t is a, we have:

$$\delta(q_1, a, q_2) = 1 \Rightarrow \mu^{t+1}(q_2) = 1.0,$$

where $\delta(q_1, a, q_2)$ is the transition weight from state q_1 to state q_2 upon receiving an input symbol a.

This observation means that a state whose activation level is as low as 0.01 ($\mu^t(q_1) = 0.01$) is able to cause its successor state to be fully activated ($\mu^{t+1}(q_2) = 1.0$). Obviously, such a membership assignment that does not consider the level of acti-

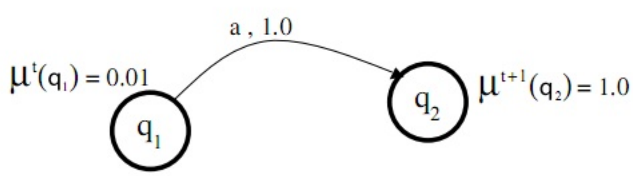


Figure 3.5: Transition-Based Membership Assignment [2]

vation of the predecessor state is not always suitable.

Therefore, the definition of the transition function in fuzzy automata has been generalized [2], to incorporate both the current state activation level and the transition weight in one loop [2]. Hence, the augmented transition function is defined as follows:

$\widetilde{\delta} : (Q \times [0,1]) \times \Sigma \times Q \overset{F_1(\mu,\delta)}{\rightarrow} [0,1]$

where $\widetilde{\delta}$ maps the active state (reached from its predecessor) to the fuzzy interval $[0,1]$ via function $F_1(\mu, \delta)$, which is called the membership assignment function, and is defined as:

$F_1(\mu, \delta) : [0,1] \times [0,1] \rightarrow [0,1]$

where $F_1(\mu, \delta)$ takes as input two parameters:

- μ: the membership grade of the predecessor state
- δ: the weight of a transition

Following this definition, the process of transitioning from state q_i to q_j upon an input a_k can be represented as:

$\mu^{t+1}(q_j) = \widetilde{\delta}((q_i, \mu^t(q_i)), a_k, q_j) = F_1(\mu^t(q_i), \delta(q_i, a_k, q_j))$

This means that the activation level of the state q_j at time $t+1$ is computed by function F_1 using both the activation level of the predecessor state q_i at time t and the weight of the corresponding transition. The function $F_1(\mu, \delta)$ is not unique, and can be modelled with several operations; the optimal choice, however, depends on the current application at hand. The most well-known in the literature, though, is the T-norm operation, which is mostly denoted by the minimum operation.

The augmented transition function $\widetilde{\delta}$ enables the finite state automaton with a kind of memory to its previous state upon transitioning to a successor one. The activation level of the predecessor state is memorized and used by the augmented transition function $\widetilde{\delta}$ upon receiving an input and completing a transition.

Multi-Membership Resolution

The second issue to be addressed is that of multi-membership, which is something inherent to the fuzzy state automaton and happens due to its fuzzy nature. It shows up under almost any situation, where multiple transitions lead to the same successor state, and therefore should be carefully addressed and properly resolved. The cruciality of this issue and the necessity of such a resolution can be inferred from the following reasons:

- fuzzy based systems usually produce a final crisp output after defuzzifying the aggregated final fuzzy set. The same concept applies to the fuzzy state automaton, in which the final multi-membership active state necessitates reduction to a single membership value that represent the final activation level of the state; hence outlining the importance of such resolution.

- even if the multi-membership active state is not final, in some applications, intermediate states also require the assignment of a single activation value. A need thus arises in systems exposed to a continuous flow of input symbols where intermediate actions need to take place based on the activation level of an active state.

- the set of membership values of successor states can be computed for each membership value of the predecessor state. This however, significantly increases the system complexity and floods the system with unnecessary information, especially in the case of a closed loop system or when the system is exposed to a large continuous flow of input symbols. This makes tracing the continuous operation of the fuzzy state automaton very difficult if not impossible.

Therefore, a single value that represents the state activation level is a necessity. This value can be then used at any stage to compute the membership value of the successor states. The problem of multi-membership is resolved by introducing another function F_2. F_2 is called the multi-membership resolution function, and is defined as [2]:

$F_2 : [0, 1]^* \to [0, 1]$

Therefore, the combination of the operations of functions F_1 and F_2 on a multi-membership state q_m leads to the multi-membership resolution algorithm [2]. The algorithm states that when several simultaneous transitions lead to the same successor active state q_m at time $t + 1$, a unified membership value or state activation value is computed as follows:

1. for each transition, the membership assignment function F_1 computes the membership grade of the successor state q_m, via the augmented transition function $\widetilde{\delta}$, and together with the state transition weight $\delta(q_i, a_k, q_m)$ and the membership value of the corresponding predecessor state q_i. The computed membership value is called v_i. This is shown in equation 3.6.

$$v_i = \widetilde{\delta}((q_i, \mu^t(q_i)), a_k, q_m) = F_1(\mu^t(q_i), \delta(q_i, a_k, q_m)) \qquad (3.6)$$

2. multiple transitions can lead to the same successor state, and the corresponding membership values are not necessarily equal. Therefore, the multi-membership resolution function F_2 further processes the set of computed membership values corresponding to each active state.

3. The result produced by F_2 is assigned as the activation level of the active state q_m, as shown in equation 3.7.

$$\mu^{t+1}(q_m) = F_2^{i=1,..,n}[v_i] = F_2^{i=1,..,n}[F_1(\mu^t(q_i), \delta(q_i, a_k, q_m))] \tag{3.7}$$

   where, n is the number of simultaneous transitions from state q_i to state q_m, $\delta(q_i, a_k, q_m)$ is the weight of the transition from state q_i to state q_m upon input a_k, $\mu^t(q_i)$ is the membership value of q_i at time t, and $\mu^{t+1}(q_m)$ is the final membership value of q_m at time $t+1$.

Similar to F_1, the choice of F_2 is not unique, and a good selection of such operation depends on the current application at hand. The most well-known in the literature though, is the S-norm operator, where the maximum operation is often used.

Output Mapping

In most applications and systems, obtaining or generating the final state activation levels is an intermediate result before producing some kind of final output or decision. In fuzzy clustering for example, we may have several active final states, each with its own activation level that represents the degree of membership of an element in a specific cluster - hence, we need to attribute output values to the states of fuzzy state automaton. However, there is no one unique explicit way of producing an output, as this can be specific to the application at hand. Therefore, this mapping should be carefully addressed to best suit the nature of the fuzzy problem.

3.2.2 Human Trust in Automation Fuzzy Model

Trust is fuzzy in nature, and many factors contribute in building up this phenomenon, for example fault size, productivity, situation and context awareness, and of course previous state of trust. At the same time, many factors also play a major role in determining each of the previously mentioned factors. For example, fault size is highly affected by the frequency of mistakes, the severity of those mistakes, and the ability of the robot and the system to recover from those mistakes;

these all play a major role in determining the fault size factor. The same applies to the productivity factor, which is highly affected by successful task completion, utility, and sophistication of the task being completed. Awareness, on another side, is thought to be determined by the robot's awareness of the human operator and his/her capabilities, the robot's awareness of its own capabilities and limitations, and the robot's awareness of the task context. From here comes our idea of building a two-level temporal fuzzy system to model such a phenomenon. In the first stage, fault size, productivity and awareness are inferred using Mamdani fuzzy inferencing models, and in the second stage, and since trust evolution is temporal, and highly related to its previous state, a finite fuzzy state machine is then implemented. The

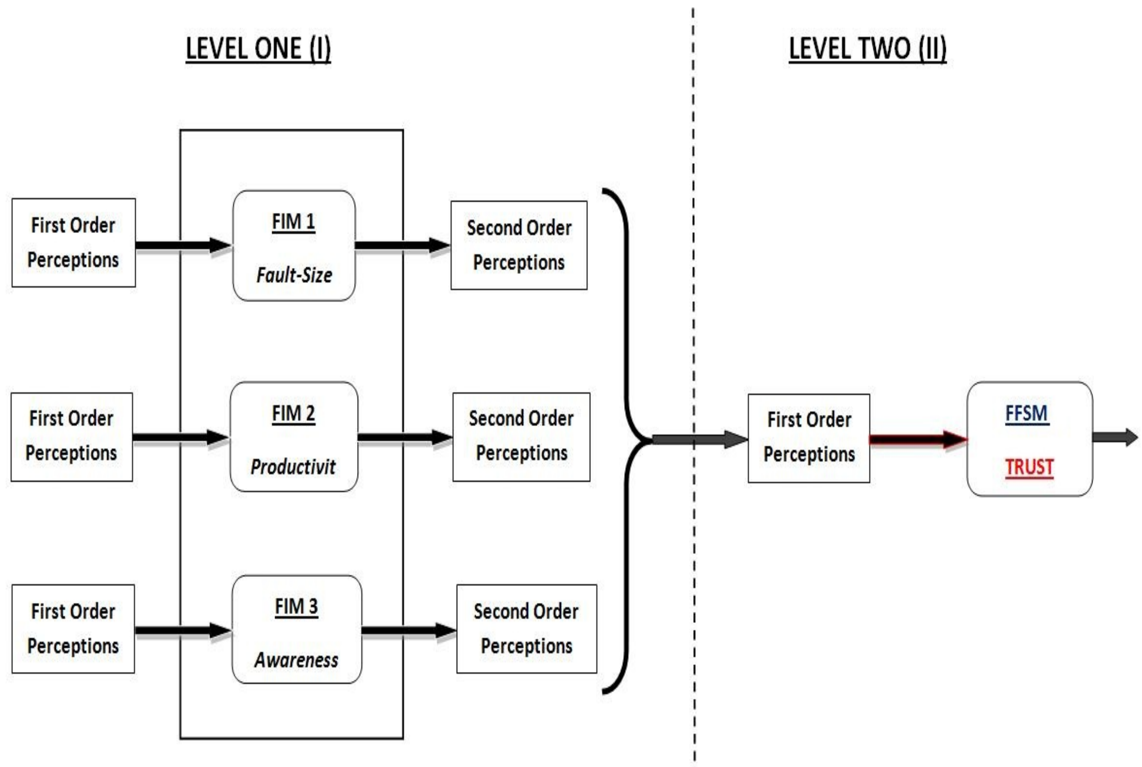


Figure 3.6: Overall Architecture of the Proposed Trust Model

overall proposed architecture shown in Figure 3.6 reduces the system's complexity and the size of the knowledge base, as will be addressed shortly.

In the following, we start by describing the FFSM model (level II) of the system, followed by level one (I).

Level Two (II)

The trust phenomenon is fuzzy by nature, and determining it depends on many factors, but among them is time and the previous state - therefore, modelling this phenomenon using only fuzzy logic might not give the anticipated results. As a result, a state machine is required, and hence, a finite fuzzy state machine that combines advantages of both fuzzy logic and the state machine is proposed. However, the larger the number of inputs to this model, the more complicated the knowledge base becomes, as the number of rules increases exponentially with the number of inputs. Hence, identifying the crucial factors that contribute largely to building up trust becomes crucial - however, the large number of factors that affect the evolution of trust makes this task very hard.

One important observation is that such factors can be grouped under a few categories. From here comes the idea of building a two-level framework to identify and estimate the level of trust. The second level relates trust to some **second-order perceptions** that contribute directly and largely into the current level of trust. Second-order perceptions are perceptions that are explained using some lower-order perceptions, or first-order perceptions. For example, in the statement *''the temperature is high, and the humidity is high, therefore the room is uncomfortable''*, the statement *''the room is uncomfortable''* is a second-order perception which is explained with the two first-order perceptions: *''the temperature is high''*, and *''the humidity is high''*. This level (Level II) is implemented using a finite fuzzy state machine that takes three inputs (which are the second-order perceptions): fault size, productivity, and awareness, and outputs the corresponding level of trust, based on the previous state of trust and the current perceived inputs (inferred from level I as will be discussed shortly). Fault size, productivity, and awareness are modelled using three membership functions: low, medium, and high. Five membership functions are used to model trust, thus projecting a smoother and more accurate switch between states. Therefore, five states are needed: very low, low, medium, high, and very high. As such, the number of required rule becomes: 3*3*3*5 = 135 rules. Figure 3.7 describes a *generic* structure of the proposed FFSM.

Table 3.1, and tables A.1, A.2, A.3, and A.4 presented in appendix A illustrate the proposed 135 corresponding rules of this FFSM. Each table corresponds to a specific current state. Table 3.1, for instance, corresponds to the case where the current active state is *very low*. Each row corresponds to a rule, which can be formulated by aggregating the input variables with an **AND** operator. Therefore, the first row in table 3.1 translates to the following rule:

Table 3.1: Current Trust State: Very Low

Rule	**Var1: FaultSize**	**Var2: Productivity**	**Var 3: Awareness**	**Output: Trust**
1	Low	Low	Low	Very Low
2	Low	Low	Medium	Very Low
3	Low	Low	High	Low
4	Low	Medium	Low	Very Low
5	Low	Medium	Medium	Low
6	Low	Medium	High	Low
7	Low	High	Low	Low
8	Low	High	Medium	Low
9	Low	High	High	Low
10	Medium	Low	Low	Very Low
11	Medium	Low	Medium	Very Low
12	Medium	Low	High	Very low
13	Medium	Medium	Low	Very Low
14	Medium	Medium	Medium	Low
15	Medium	Medium	High	Low
16	Medium	High	Low	Very Low
17	Medium	High	Medium	Low
18	Medium	High	High	Low
19	High	Low	Low	Very Low
20	High	Low	Medium	Very Low
21	High	Low	High	Very Low
22	High	Medium	Low	Very Low
23	High	Medium	Medium	Very Low
24	High	Medium	High	Very Low
25	High	High	Low	Very Low
26	High	High	Medium	Very Low
27	High	High	High	Low

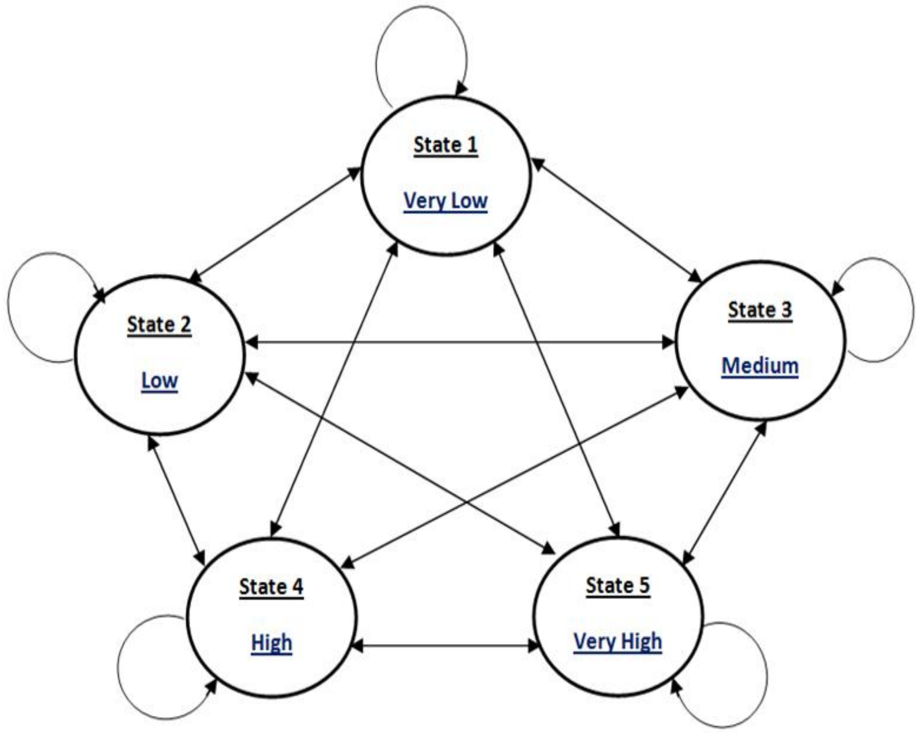


Figure 3.7: A **Generic** Model of the Trust Fuzzy Temporal States

Rule 1: **IF** q(very low)[t-1], AND fault size is low, **AND** productivity is low, **AND** awareness is low, **THEN** q(very low)[t].

The rule states that if the current state corresponds to *very low* trust, and the current inputs indicate that the fault size, productivity, and awareness levels are *low*, then the next state will correspond to a *very low* trust.

The components of the proposed FFSM are:

- states: every state represents a phase in the temporal evolution of human trust. The system has five states representing trust: very low, low, medium, high, and very high. Initially at time zero, when the human starts interacting with the robot, the trust is usually based on major factors:
  - intrinsic human trust: every human being has a different level of initial intrinsic trust. This value varies from one human being to another,

depending on many psychological and sociological factors.

  – reputation: people tend to be biased by what they hear about machines. Therefore, a robot's reputation is very important in determining the basic initial level of trust it will be granted. Robots with good reputations will be more trusted when first activated, and vice versa.

  In this work, we assume the initial state to be medium, and that the human trust in the machine is not very much initially affected by faith, affection, or reputation.

- input alphabet (IA): natural phenomena are usually described in terms of signals and variables. Assuming a scenario with only one input variable x, that is represented with m fuzzy sets and membership functions $A_1, A_2, ..., A_m$, it is possible to create the temporal description of the signal in the form of a fuzzy input alphabet $(\mu_1/A_1, \mu_2/A_2, ..., \mu_m/A_m)$. The proposed FFSM has 3 input variables: fault size, productivity, and awareness; each is represented with three membership functions: low, medium, and high. Using these linguistic terms, we can create an input fuzzy alphabet as follows:

  fault-size(μ_{Low}/Low, μ_{Med}/Med, $\mu_{High}/High$),
  productivity(μ_{Low}/Low, μ_{Med}/Med, $\mu_{High}/High$),
  awareness(μ_{Low}/Low, μ_{Med}/Med, $\mu_{High}/High$)

  This alphabet allows us to create a description of the perceptions evolution in time. An example of an instance of such an input alphabet at time t, could be modelled as:

  fault-size(0.8/Low, 0.2/Med, 0.0/$High$),
  productivity(0.0/Low, 0.6/Med, 0.4/$High$),
  awareness(0.3/Low, 0.7/Med, 0/$High$)

- output alphabet (OA): the purpose of the proposed finite fuzzy state machine is to monitor the level of human trust in a robot when performing a task. Five states, as described earlier, are used to model the temporal evaluation of trust, so the output alphabet would be:

  Trust($\mu_{VLow}/VLow$, μ_{Low}/Low, μ_{Med}/Med, $\mu_{High}/High$, $\mu_{VHigh}/VHigh$)

  The output string is obtained by applying the FFSM to the input string.

- transition function: this function is defined as $Q[t] = f(IA, Q[t-1])$ where current state activation is determined based upon current input and the previous system state. The function behaviour is governed by a set of propositions in the form of *if-then* fuzzy rules. Five states are used in this FFSM model: very low, low, medium, high, and very high. The number of required rules is 135 (as previously discussed).

- output function: this function combines all activation levels of current active states, and comes up with one representative output. Obtaining or generating the output alphabet is an intermediate result before obtaining the crisp final value of trust based on the generated state membership grades or state activation levels. For this reason, we follow in this work the same structure defined by Sugeno-type fuzzy inference models, and map the states (very low, low, medium, high, very high) to the following zero-order consequents $(0.1, 0.3, 0.5, 0.7, 0.9)$. The crisp overall consequent is then generated by aggregating the qualified crisp output of each rule using the weighted average method, as described in the equation 3.5.

Following the same description of state assignment and multi-transition resolution presented in earlier sections, we further explain the inference mechanism and state activation assignment in our proposed finite fuzzy automaton. Let Σ be a finite set of symbols (input alphabet), and Σ^* be the set of all possible combinations over Σ (cross product). For instance, assume that the three input variables, fault size, awareness, and productivity, are represented with the symbols A, B, and C respectively, where each variable is represented with three fuzzy sets.

$\Sigma = \{A, B, C\}$, where $A = \{A_1, A_2, A_3\}$, $B = \{B_1, B_2, B_3\}$, and $C = \{C_1, C_2, C_3\}$

Therefore,

$\Sigma^* = A \times B \times C = \{(A_1, B_1, C_1), (A_1, B_1, C_2), .., (A_3, B_3, C_3)\}$

$\Sigma^* = \{x_1, x_2, .., x_{27}\}$, where $x_1 = (A_1, B_1, C_1)$, etc.

Thus,

$|\Sigma^*| = |A| \times |B| \times |C| = 3 \times 3 \times 3 = 27$ represents the cardinality of Σ^*, and the size of the knowledge base representing each state.

Let α be the fuzzy description of a new system input:

$\alpha = \{\mu_A, \mu_B, \mu_C\}$

$\alpha = \{(\mu_{A_1}, \mu_{A_2}, \mu_{A_3}), (\mu_{B_1}, \mu_{B_2}, \mu_{B_3}), (\mu_{C_1}, \mu_{C_2}, \mu_{C_3})\}$

Therefore, following the decription of Σ^*, α^* can be formulated as:

$\alpha^* = \{(\mu_{A_1} \otimes \mu_{B_1} \otimes \mu_{C_1}), (\mu_{A_1} \otimes \mu_{B_1} \otimes \mu_{C_2}), .., (\mu_{A_3} \otimes \mu_{B_3} \otimes \mu_{C_3})\}$

$\alpha^* = \{\delta_1, \delta_2, .., \delta_{3\times3\times3}\}$, where $\delta_1 = \mu_{A_1} \otimes \mu_{B_1} \otimes \mu_{C_1}$, etc.

where the operator $\otimes$ denotes the T-norm operation used in this work.

We now introduce the function that represents the transition from fuzzy state $\widetilde{P}$ to a new state $\widehat{\mu}(\widetilde{P}, x_i)$ induced by the symbol x_i contained in the fuzzy description Σ^*. Let Q be the set of possible fuzzy states. The function $\widehat{\mu} : Q \times \Sigma \rightarrow Q$ is defined as shown in equation 3.8, where $\otimes$ and $\oplus$ denote the T-norm and S-norm operators respectively [128].

$$\widehat{\mu}(\widetilde{P}, x_i) = \{(p, \mu) | \mu = \oplus_{q \in Q}(\mu(q, p, x_i) \otimes \mu(q)), q \in Q\} \tag{3.8}$$

where $\mu(q, p, x_i) = \delta_i$, if such a transition from state q to state p exists, and 0 otherwise. Equation 3.8 is applied repeatedly to all variables x_i with corresponding firing strength $\delta_i \neq 0$.

Therefore, the states' activation levels are calculated as shown in equation 3.8. The equation states that the T-norm operation is used to determine the firing strength of each fired rule. Then, the S-norm operation is used to choose the firing strength that corresponds to the most dominant rule that reaches a specific state, and then noted as the state activation level. This inference mechanism is followed in our implementation.

In this work, Max-Min operation is used for determining the state activation level. The "AND"minimum T-norm operation is used to determine the firing strength of each fired rule, and the maximum firing strength that corresponds to the most dominant rule that reaches a specific state is regarded as the state activation level.

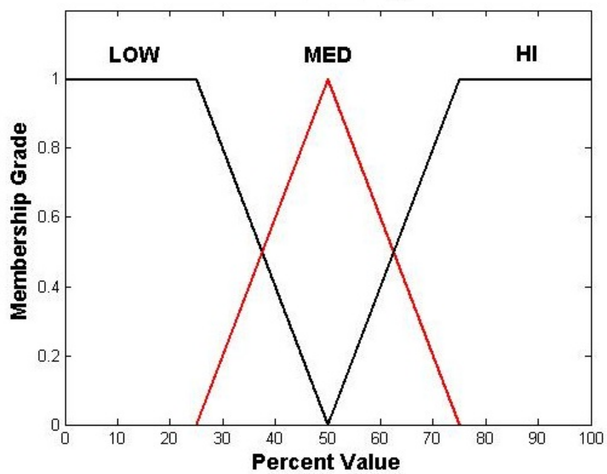


Figure 3.8: Low, Medium, and High Membership Functions

Level One (I)

As discussed earlier, level II is the direct connection between trust and the most influential second-order perceptions that contribute directly to its temporal evolution. A finite fuzzy temporal inferencing model is used to model this relationship as trust highly depends on its previous state. However, the second-order perceptions inputted to the FFSM are themselves explained with lower order perceptions, and hence should be properly modelled. Fault size (FS), for example, is highly determined by the three lower order factors: fault frequency (FF), fault cruciality (FC), and fault recovery (FR). Such factors are not temporal, hence no FFSM is needed, and therefore, a fuzzy Mamdani inferencing model is proposed to infer the resultant fault size depending on the previously mentioned factors. Each of those factors is fuzzy in nature, therefore each is represented with three membership functions: low, medium, and high. Therefore, twenty-seven rules are needed. Figure 3.8 shows the chosen membership functions.

The same applies to awareness. Three inputs are regarded to largely determine the value of this factor: machine awareness of its capabilities (MA), context awareness of the task (CA), and machine awareness of the human operator's availability and cognitive and physical abilities and limitations (HA). Each factor is represented with three membership functions: low, medium, and high. Therefore, twenty-seven rules are required to represent the knowledge base of this model. Finally, as for productivity, two inputs mainly determine the value of this factor: task/goal suc-

Table 3.2: FIM3: Productivity

Rule Nb	**Var1: Task Completion**	**Var2: Task sophistication**	**Output: Productivity**
1	Low	Low	Low
2	Low	Medium	Low
3	Low	High	Medium
4	Medium	Low	Medium
5	Medium	Medium	Medium
6	Medium	High	Medium
7	High	Low	Medium
8	High	Medium	High
9	High	High	High

cessfulness and completion (TC), and task complexity and sophistication (TS). Each factor is represented with three membership functions: low, medium, and high. Therefore, nine rules are used to represent the knowledge base of this model.

Tables B.1 and B.2 shown in appendix B present the knowledge base of both the fault size and awareness proposed fuzzy inferencing models. Table 3.2 presents the knowledge base of the productivity model. A Mamdani fuzzy inference model is used, where the output in each of the above-mentioned three cases is also represented with the same three membership functions: low, medium, and high. Figure 3.9 shows the fuzzy surface generated for the productivity fuzzy inferencing model.

The idea behind building a two-level architecture to estimate the trust factor is important as it dramatically decreases the complexity of the system and the size of the knowledge base; as a matter of fact, simple calculations show that without this two-layer architecture, and using only a FFSM to model this factor, keeping the same number of inputs and membership functions, $5 * 3^8 = 32,805$ rules would have been needed to implement such a knowledge base, which is impractical and very difficult to implement. However, using the proposed two-level architecture, 135 rules are needed to represent the second level, and sixty-three rules to implement the three fuzzy inference models proposed in level one.

Summary of System Interconnection

The proposed trust evaluation model is a two-level system, where level II uses a finite fuzzy state machine to infer the trust state, which can take one or more of five states, very low, low, medium, high, and very high, based on some three essential

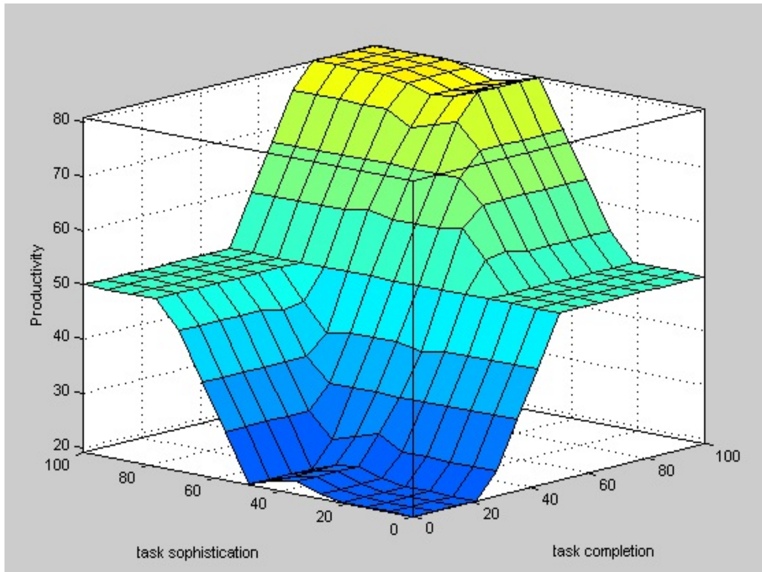


Figure 3.9: Fuzzy Surface for Productivity Fuzzy Inference

second-order perceptions, fault size, productivity, and awareness, which can be as well explained using some first-order perceptions, and this is done in level I. Level II uses a finite state machine as trust evolves with time, and depends not only on some new inputs, but also on its previous state. Level I, on the other hand, comprises three Mamdani fuzzy inference models, that explains or infers the previously stated three perceptions based on first-order perceptions and some knowledge base. Fault size, for example, is inferred using a fuzzy model that takes as inputs the fault frequency, the fault cruciality, and the fault recovery, which, all combined with some predefined knowledge base, is able to infer the fault size. A simplistic map for this architecture is shown in Figure 3.10.

3.3 Proposed Human Reliability Fuzzy Model

Much research that addresses human performance in systems is presented in the literature [129], [130], [131], [132]. Several studies show that human factors are responsible for 20% to 90% of the failures in many systems. For example, according to [129], 70% to 90% of system failures are directly or indirectly related to human error. Finnegan [130] also found that more than 20% of fossil-fuel power-plant system failures were directly related to human wrong actions, as in incorrect procedures, accidental operations, maintenance errors, and misuse of instruments

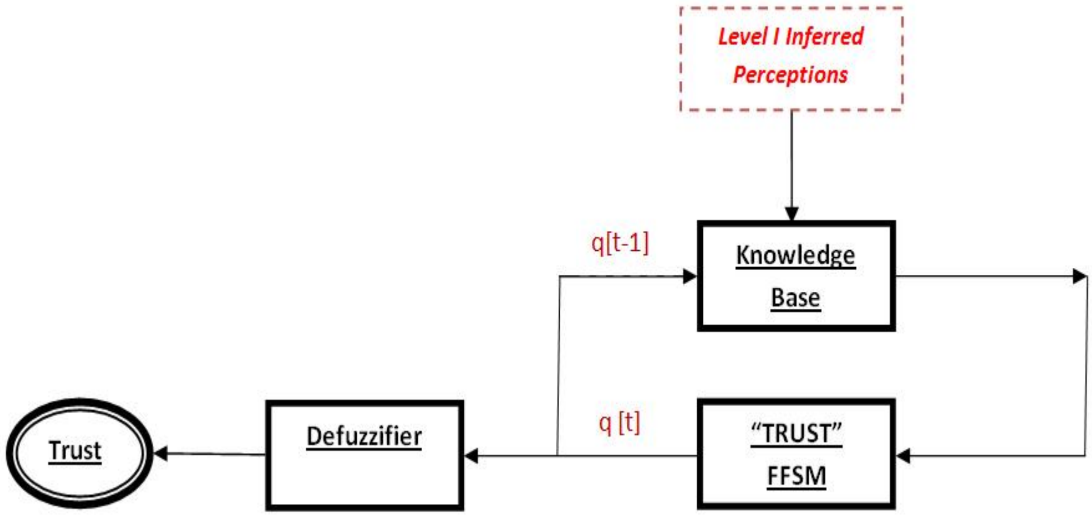


Figure 3.10: Proposed Trust Model Diagram

[133]. In parallel, Santoni et al. [132] study the performance of automated systems which concentrate information and decision making on the hands of human operators, who under time pressure are subject to high cognitive loads. In the electricity industry, it was found that 20% of system failures are related to human errors. Another related work reviews and analyzes unmanned aircraft (UA) accident data that were collected from the U.S. army, navy, and air force. The study shows that 21% to 68% of the accidents were related to human involvement [131]. In his book, Dhillon [134] also states that human operator errors account for more than 50% of the overall technical medical equipment problems. A study of anesthetic incidents revealed that between 70% to 82% of the incidents were due to human errors. The center for devices and radiological health (CDRH) of the food and drug administration (FDA) also reports that human error accounts for 60% of all medical device related deaths or injuries.

Therefore, one can notice the crucial importance of modelling this factor when building an effective and a generalized robotic metric framework. Lots of researchers addressed the problem of modelling human reliability: fuzzy logic and probabilistic approaches were used for this purpose [135]. In this work we believe that modelling human reliability should be fuzzy - and since this factor is highly related to time, and to its previous state, a FFSM is then required. The same description that was presented for level II in trust modelling is used for modelling the human reliability factor.

Many factors affect humans and their performance; some are psychological, physiological, sociological, while others may be related to environmental and external factors such as weather situation, pressure, and temperature. In this work, however, we will only be focusing on those factors that arise from interacting directly with the robot or the machine while the team is working to accomplish a task. Therefore, three input variables are considered: (1) number of subtasks being simultaneously perceived (NS), (2) mental workload required during task completion (MW), and (3) external/internal burden (EB). EB is an estimate of many factors such as external temperature, social problems, pressure, and/or stress, among others.

Although the HR factor is crucial and important, we will not further explore it, as this would require further in depth studies in the disciplines of sociology, psychology, and physiology, which goes beyond the scope of this work. Therefore, we present a preliminary suggested set of rules to model the knowledge base of this phenomenon, with the hope that this work represents a building block or a further step toward better modelling of this factor in some related upcoming future work. Three membership functions are used to model each variable: low, medium, and

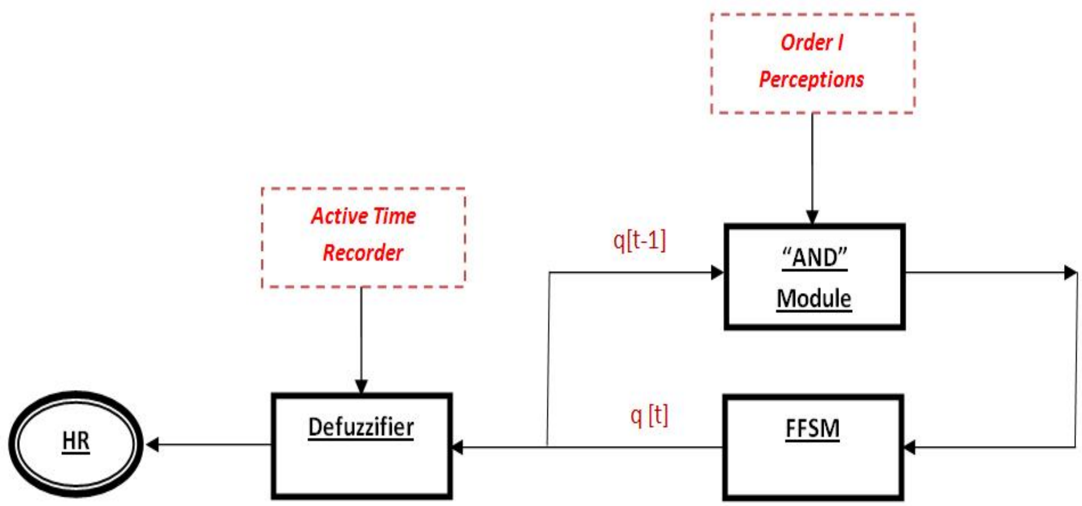


Figure 3.11: Proposed Human Reliability Model Diagram

high. The human reliability factor is modelled using five membership functions: very low, low, medium, high, and very high. Hence, five states, and a total of 135 rules are required to model such phenomenon. Table 3.3, and tables C.1, C.2, C.3, and C.4 in appendix C, present such a preliminary suggested set of rules. The same rule structure and composition, as described earlier, is used in modelling the knowledge base of the human reliability factor.

However, one important factor that we should keep in mind is time. Although the above-mentioned three variables play a remarkable role in determining the human reliability, interaction time, or time while being active, is also crucial, as human performance degrades with time even when the task is simple and does not impose much cognitive and physical load on the human operator. Figure 3.11 shows the overall structure of the proposed human reliability module, in which the output function, also denoted as the defuzzifier, maps the states (very low, low, medium, high, very high) to the following consequents (0.1−a×t,0.3−a×t,0.5−a×t,0.7−a×t,0.9−a×t), where a is a subjective constant that varies from one human being to another and represents the natural human degradation factor, and t represents the time factor.

The crisp overall consequent is then generated by aggregating the qualified crisp output of each rule using the weighted average method, as described in equation 3.5. Introducing the time factor means that even when the human reliability is still in the state "very high"for example, this does not mean that it can stay in that state indefinitely, as the consequent that corresponds to the state "very high"will decline with time, hence accommodating for the natural human time-related degradation aside from the human-robot interaction context and any other external factors. In this work, we assume a 5% natural degradation factor value per time unit. Note that we chose to model the natural degradation linearly; exponential modelling could also be used, but since a small value of a is usually the case, both linear and exponential representations would behave similarly, and hence the linear form is chosen for its simplicity.

3.4 Metrics Generalization Models for Multi-Robot Systems

So far throughout this work, our focus has been drawn toward one-robot systems, where a human user is interacting with one robotic agent toward achieving some tasks. However, an important issue that Olsen and Goodrich, among many other researchers, ignored, is when the robotic system is composed of more than one robot. For example, addressing the RAD metric as described by Olsen, does it

Table 3.3: Current Human Reliability State: Very High

Rule Number	**Var1: Nb of Subtasks**	**Var2: Mental Workload**	**Var3: External Burden**	**Output: Human Reliability**
1	Low	Low	Low	Very High
2	Low	Low	Medium	Very High
3	Low	Low	High	Very High
4	Low	Medium	Low	Very High
5	Low	Medium	Medium	Very High
6	Low	Medium	High	High
7	Low	High	Low	Very High
8	Low	High	Medium	High
9	Low	High	High	High
10	Medium	Low	Low	Very High
11	Medium	Low	Medium	Very High
12	Medium	Low	High	High
13	Medium	Medium	Low	Very High
14	Medium	Medium	Medium	High
15	Medium	Medium	High	High
16	Medium	High	Low	High
17	Medium	High	Medium	High
18	Medium	High	High	High
19	High	Low	Low	Very High
20	High	Low	Medium	High
21	High	Low	High	High
22	High	Medium	Low	High
23	High	Medium	Medium	High
24	High	Medium	High	High
25	High	High	Low	High
26	High	High	Medium	High
27	High	High	High	Medium

mean that a two-robot system will have twice the RAD compared to a one-robot system? If so, can we still use it as an indicator to how well the system is performing? Of course not, because if we just ignore the fact that these two robots are a part of a bigger system, working toward achieving a bigger goal, then we are restricting this metric from being used as a generic indicator of the performance of the overall human-robot team.

Therefore, the performance metric should evaluate the overall performance of the whole system, and be a good indicator of how well the team is performing (the human user and the robotic system). Hence, a special consideration of multi-robot systems should be also addressed. Several cases emerge from this fact, as these robots can have sequential or parallel ways of executing their tasks, and with varying levels of dependency, therefore each case should be individually addressed. In this chapter, then, we further extend and generalize our proposed generic metric to accommodate each of the previously mentioned scenarios. Two-robot systems are first addressed, and a generalized conclusion is then extended to N-robot systems.

3.4.1 Sequential Execution of Tasks

In this scenario, two robots are cooperating with the guidance of a human user to accomplish a specific task. Doing so, only one of the two robots is active at one time, and the other robot is idle waiting for the first robot to finish its task so the human user can instruct it about the next one. In this scenario and since only one robot is active at some point of time, the system FO is simply the FO of the active robot. Therefore, if we note the FO of the idle robot to be equal to zero, the system FO can be defined as the logical OR of the two robots' FO. This conclusion can be generalized to N-robot systems, as shown in equation 3.9.

$$SystemFO = FO_1 \vee FO_2 \vee ... \vee FO_N \tag{3.9}$$

where $FO(idlerobot) = 0$.

It is worth noting here that the robot FO is human reliability dependent, where the latter one is time dependent, and hence it is very important to note that the human reliability propagates between active robots, and does not simply start all over again. This will be clearly shown in the next chapter where supportive simulations are presented.

3.4.2 Parallel Execution of Tasks

Independent Execution of Tasks

In this scenario, two robots are cooperating with the guidance of a human user to finish a specific task. Doing so, the two robots are active at the same point executing independent tasks in parallel. In this scenario, tasks conducted by different robots are independent, and hence, dependency-related issues can be ignored. An important point to note though is that these robots are active simultaneously, and hence, it is a valid statement to say that the system complexity, the number of subtasks, as well as the mental workload are higher. So the human reliability is more probable to decrease at a faster rate compared to a one-robot system. As for the overall system FO, the average of the two robots' FO is usually indicative of the system FO, but since not all robots contribute equally to the overall task completion, a weighted average method is applied to find the overall system FO. This can be generalized to N-robot systems, as shown in equation 3.10, where W_i is the percent contribution of the i_{th} robot toward the final goal completion.

$$SystemFO = \sum_{i=1}^{N} W_i FO_i, where \sum_{i=1}^{N} W_i = 1 \tag{3.10}$$

Dependent Execution of Tasks

This scenario is the most complicated compared to the previous ones. First, we consider two robots that are working with the guidance of a human user toward achieving some specific tasks. The two robots are active at same point executing dependent tasks in parallel. The system FO can be approximated with the weighted average method of all individual robots' FO when tasks conducted by different robots are independent, but in this case, dependency-related issues should be addressed. In such a case, the system FO will fall somewhere between this weighted average value and the smallest robot FO (the one with the poorest performance), as such task dependency will most likely restrict the better robot, by making it wait for the slower robot to finish executing its task, for instance. In fact, when total (100%) task dependency is encountered, the system FO is forced to follow the smallest robot FO; the less dependency exists, the closer the system FO will be to the weighted average (the task independent scenario). Therefore, the system FO is defined as shown in equation 3.11, where d represents the percent task dependency between robots one and two, and W_i is the percent contribution of the i_{th} robot toward the final goal completion.

$$SystemFO = d * min(FO_1, FO_2) + (1 - d) * \sum_{i=1}^{2} W_i FO_i \tag{3.11}$$

Generalizing this conclusion to N-robot systems is a bit more complicated than the two-robot case. For example, for a three-robot system, inter-task dependencies between robots 1 and 2, robots 1 and 3, and robots 2 and 3 should be accounted for. Therefore, the system FO is defined as shown in equation 3.12, where $d_{i,j}$ is the task percent dependency between robots i, and j, W_i is the percent contribution of the i_{th} robot toward the final goal, and FO_i is the fan-out of the i_{th} robot.

Equation 3.12 consists of two basic parts, in which the second part (between braces) represents the *relative* weighted average FO of the i_{th} and j_{th} robots. Relative and not individual weights are used, as every two-robot system is addressed at a time as a whole subsystem. The first part of the equation (the fraction) explains how much contribution both robots make toward the final goal. When dealing with pairwise dependencies, however, $\binom{N}{2}$ possibilities arise, in which, the sum of all contribution weights corresponding to all possibilities is equal to $(N-1)$, hence explaining the division by $(N-1)$.

Ideal System FO=

$$\sum_{i<j}^{N} \frac{W_i + W_j}{N-1} * \left\{ d_{i,j} * min(FO_i, FO_j) + (1 - d_{i,j}) * (\frac{W_i}{W_i + W_j} FO_i + \frac{W_j}{W_i + W_j} FO_j) \right\} \tag{3.12}$$

The above mentioned equation states that in a four-robot system, $\binom{4}{2} = 6$ pairwise dependencies exist: (1) robots 1 and 2, (2) robots 1 and 3, (3) robots 1 and 4, (4) robots 2 and 3, (5) robots 2 and 4, (6) robots 3 and 4. Summing the contributions made by all these cases, we get $(R1+R2)+(R1+R3)+(R1+R4)+(R2+R3)+(R2+R4)+(R3+R4)$, which can be regrouped as $3*(R1+R2+R3+R4)$, where the sum of contributions made by robots $R1$, $R2$, $R3$, and $R4$ is equal to 100%, hence a division by 3 is needed. This remark can be generalized to N robots, as outlined in equation 3.12.

An important issue that arises in this scenario, though, is that inter-task dependencies might possibly affect other dependencies in the system. Meaning that the dependency between robot 1 and robot 2, might also have some negative implication and cause additional cost on the dependency between robot 2 and robot 3, and so on. Therefore, the above specified system FO can be regarded as the ideal case, and hence, the real and practical system FO is regarded to be upper bounded by the above evaluated measure, and lower bounded by the smallest robot FO. This is shown in equation 3.13.

$$min(FO_1, FO_2, ..., FO_N) < RealSystemFO < IdealSystemFO \qquad (3.13)$$

3.5 Chapter Summary

In this chapter, we present a further step toward identifying a common generic metric to assess the performance of the human-robot team. Doing so, the focus is first drawn to one-robot systems, in which a human user is collaborating with only *one* robot to achieve some tasks. We present an alternative definition of fan-out (FO) in terms of robot attention demand (RAD) and human reliability (HR). Then, we attempt to determine the true time that an operator has to dedicate to the robot. Therefore, we define the RAD as a function of both direct interaction time (DIT), and indirect interaction time (IIT), where the IIT is a direct consequence of trust, and can represent the time being spent when the robot is being neglected, but still with much of the user's attention drawn to it as a result of the operator's distrust in the machine.

We also propose a two-level fuzzy system to model the crucial human trust in automation phenomenon. In the first stage, fault size, productivity, and awareness are inferred using Mamdani fuzzy inferencing models, and in the second stage, and since trust evolution is temporal and related to its previous state, a finite fuzzy state machine is then implemented. The proposed model significantly reduces the system complexity and the size of the knowledge base by grouping perceptions into first- and second-order perceptions. Another temporal model is proposed for human reliability assessment. This model uses a finite fuzzy state machine to estimate the human reliability state; first-order Sugeno-like consequences are used for defuzzifying the active states, as human reliability degrades naturally with time even when task complexity is simple and does not impose much cognitive and physical load on the human operator. Finally, this metric is augmented to accommodate multi-robot systems. Several cases emerge, as these robots can have sequential or parallel ways of executing their tasks, and with varying levels of dependency. Hence, each case is individually considered.

Chapter 4

Preliminary Simulation Results

In the following chapter, we discuss some numerical simulation results that intuitively explain and support our proposed generic metric framework. Two main scenarios are to be addressed as described earlier: one-robot systems, and multi-robot systems.

4.1 One-Robot System

In this section, we present some simulation results related to a one-robot system, obtained when the proposed trust and human reliability models are put in action. Figure 4.1 shows the trust evolution in accordance with the temporal change of the three inferred second-order perceptions discussed earlier. An estimate of the inferred fault size, awareness, and productivity is generated randomly in order to discuss their implication on trust evolution as estimated by our proposed trust model (level II precisely). Results shown in Figure 4.1 show that at time equal zero (t = 0, which represents the time at which the human-robot interaction starts taking place), the human trust in automation is assumed to be neutral. Then the trust factor varies according to new perceptions. An important thing to note is the smoothness of change in this trust factor value, which is observed by the incremental decrease and increase in its value rather than an abrupt change. This is highly related to the fact that human trust is a step by step process, and strongly depends on its previous state. Results also show that when the fault size is low, and both awareness and productivity are high, the human trust in automation is high, and vice versa.

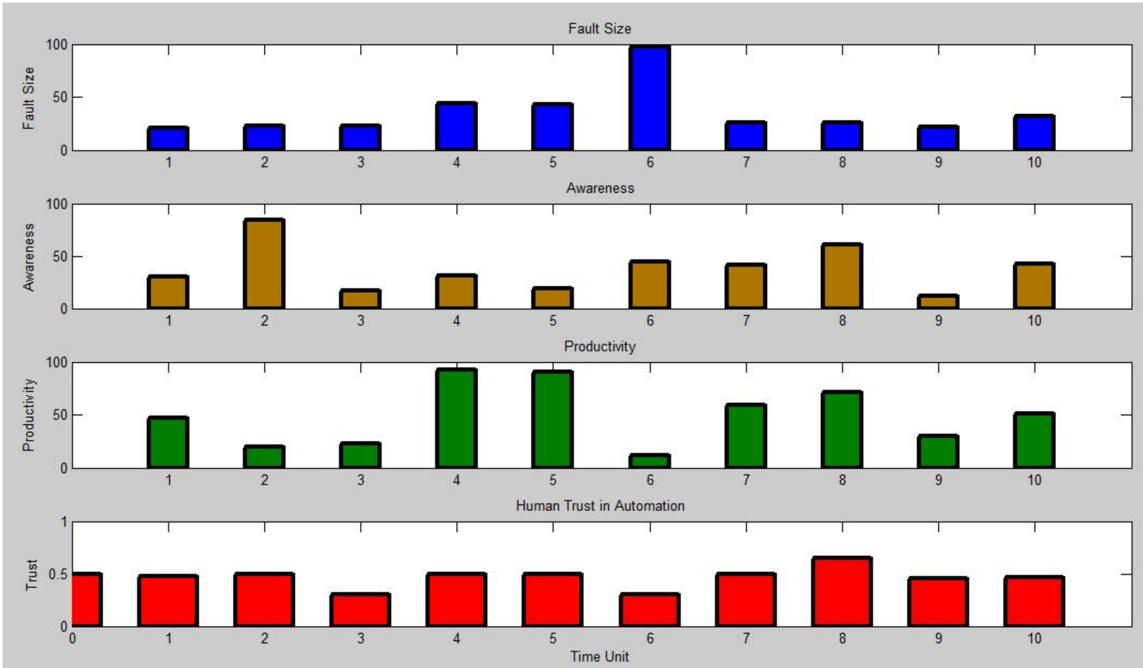


Figure 4.1: Level II Trust Simulation Results -1-

Another interesting scenario for estimating the human trust in automation is shown in Figure 4.2, where the most extreme conditions are taken into account. Initially at $t = 0$, trust is assumed to start at a medium state (50%), and then from t = 1 to t = 3, fault size is assumed to be low (10%), and both productivity and awareness are assumed to be high (90%); trust evolution is observed. Results show that trust only increased smoothly and step by step, which reflects the most intuitive trust characteristics. According to Sisodia et al. [136], developing trust is a slow process, and maintaining it is always a challenge. Notter et al. [137] also states that building trust is done in steps and over time. Trust has an organic nature, it grows through our own actions, and doesn't just pop off the assembly line. Then, starting from t = 4 to t = 7, fault size is assumed to be high (90%), where productivity and awareness are assumed to be low (10%). Trust evolution shows a relatively faster decrease in trust compared to when trust was building up, and then the decrease keeps occurring but at some lower rate. The reason for that is very intuitive and rooted to human psychology; humans tend to lose trust much faster than building it. Seeing something that conflicts with our faith about something makes us suspicious and precautious about it, and thus largely decreases our level of trust. Trust is hard to earn, and easy to lose [138].

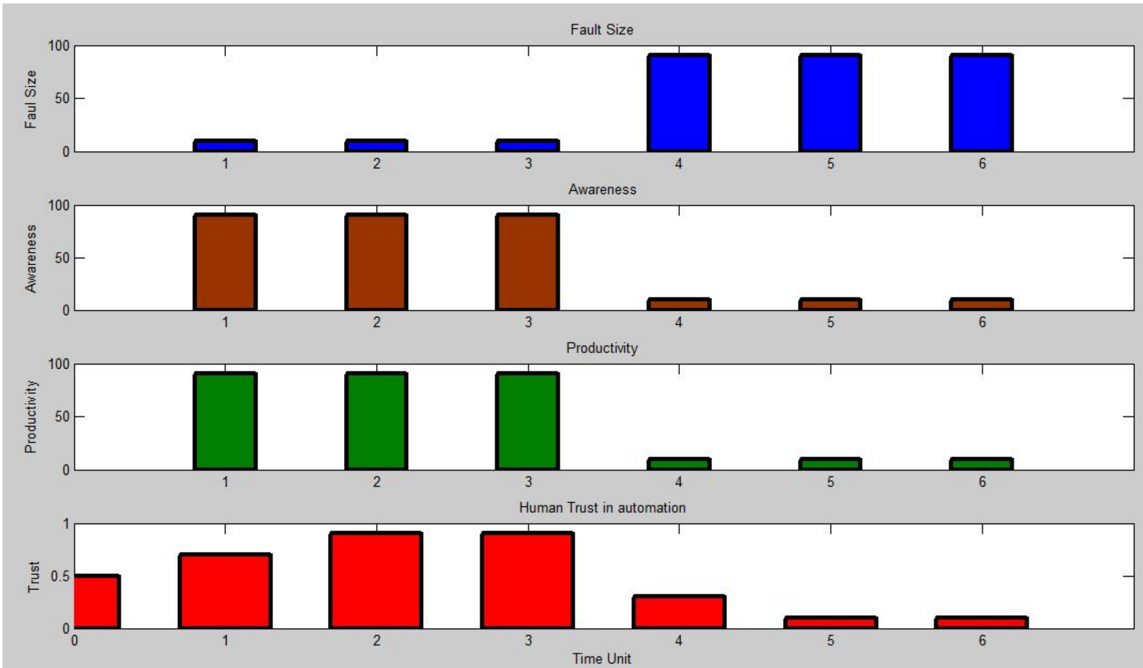


Figure 4.2: Level II Trust Simulation Results -2-

On the other side, regarding the human reliability proposed model, and in order to emphasize the importance of the proposed first-order temporal consequences when mapping the fuzzy state situation to a crisp human reliability value, we simulate the model in yet another extreme scenario, in which a human user operates in a comfortable working environment - where the number of subtasks is low, as well as the mental workload, and the external and internal burden. Such comfortable working conditions that do not impose much cognitive and physical load on the human operator are thought to keep the human reliability in its optimum state (*very high*), but not indefinitely. The natural human degradation factor, which is the reason behind choosing first-order temporal consequences instead of constant zero-order ones, addresses this fact, and shows that the human reliability degrades even in the most comfortable scenarios as interaction time increases. This result is shown in Figure 4.3. This factor is subjective and varies from one person to another. Figure 4.3 shows the human reliability natural degradation for different scenarios where the natural degradation factor is assumed to be 2%, 5%, and 8%, compared to the ideal, yet unrealistic, case where the human reliability is assumed to be constant in simple working conditions. In this work, a 5% natural degradation

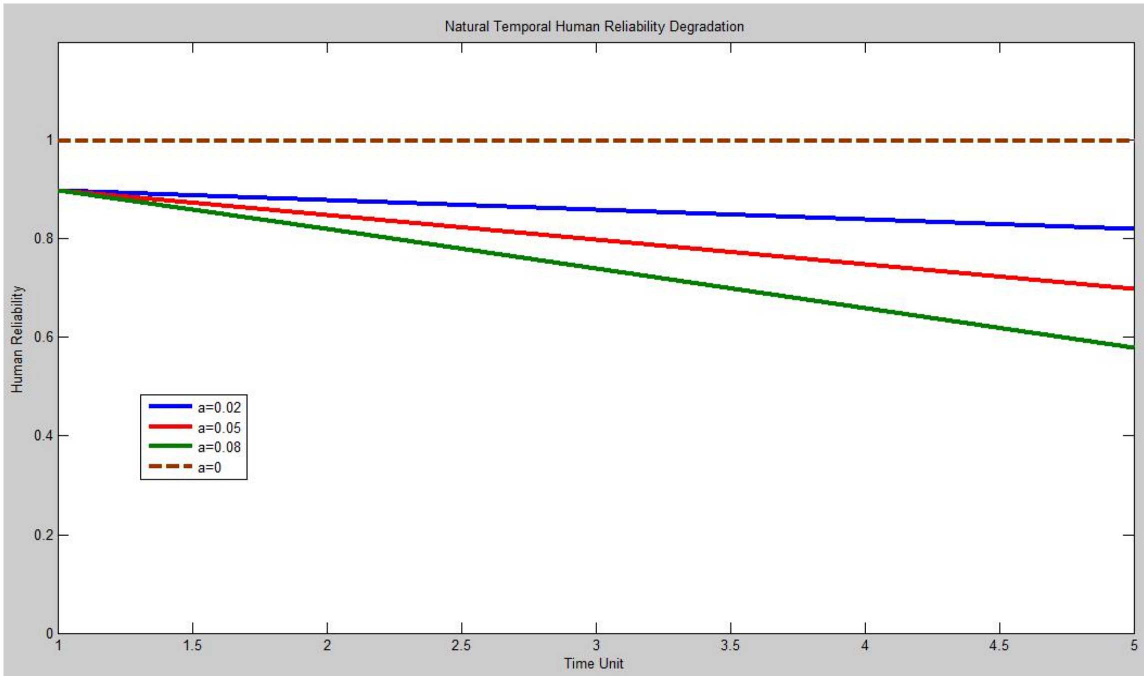


Figure 4.3: Level II Human Reliability Simulation Results -1-

factor per time unit is assumed. Note that a linear degradation model is chosen in this work. Other exponential models could be also used, but since a small value of the degradation factor a is usually the case, both linear and exponential models behave similarly; hence, the linear model is chosen for its simplicity.

Still in the human reliability assessment, Figure 4.4 shows the human reliability evolution in time, where an estimate value of the number of subtasks, mental workload, and external and internal burden, is generated randomly in order to discuss their implication on the HR evolution as estimated by our proposed human reliability model. Results, shown in Figure 4.4, show that at time equal to zero (t = 0, which represents the time at which the human-robot interaction starts taking place), the human reliability is assumed to be very high (90%). Then the HR factor varies according to new perceptions. The most important thing to notice here is the smoothness of change in this factor, which is highly related to the fact that the human reliability's current value is directly dependent on its previous state. Results also show that when the number of subtasks, mental workload, and external and internal burden are high, the human reliability tends to decrease to become really low, and vice versa when such perception values tend to decrease.

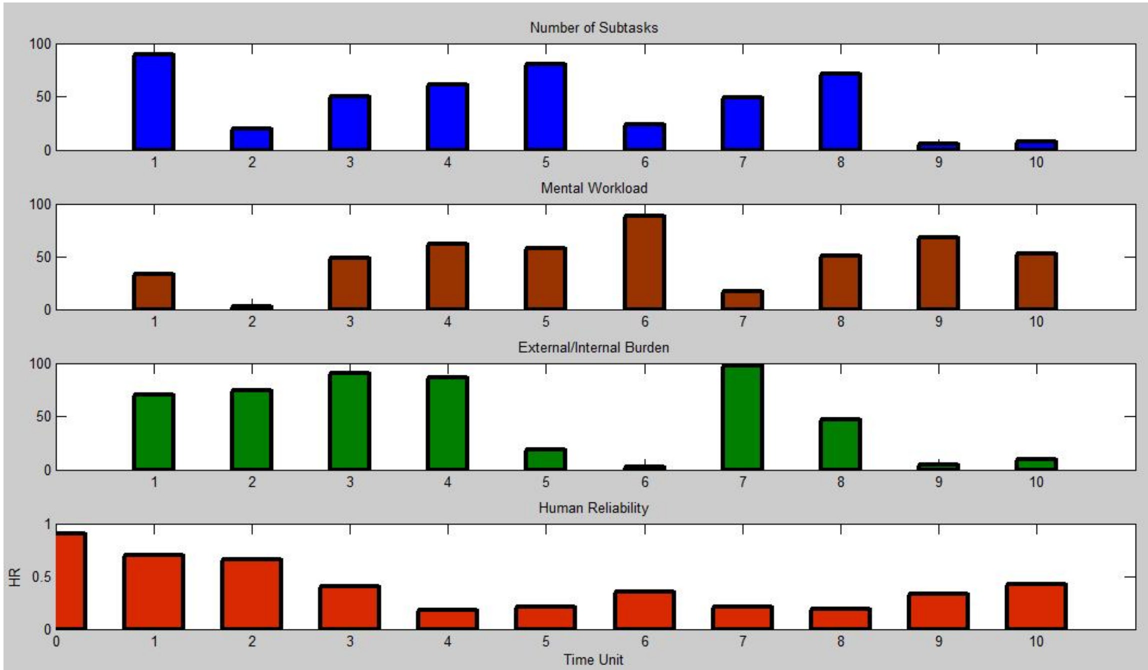


Figure 4.4: Level II Human Reliability Simulation Results -2-

Then, we study the implication of trust and human reliability as assessed on the overall proposed performance metric. Figure 4.5 shows the implication of the human trust in automation factor on both interaction time (IT) and free time (FT). Direct interaction time (DIT) is assumed to be 25% of the overall task time, during which, the human user instructs and informs the robot about the task to be completed. Figure 4.5 shows that when the human trust in automation increases, the indirect interaction time (IIT) spent monitoring the system and making sure the robot's efficiency does not drop below a specific threshold (as in getting stuck at a dead end) - thus interfering when such situation becomes highly probable - decreases. Results show that at time t = 2, when the human trust in automation is very high, the IIT is almost negligible, and hence the practical free time, which is the time during which the human operator can neglect the robot, and conduct another task or instruct another robot, becomes closer to the ideal free time (obtained using only the DIT). Results also show that at time t = 4, when the human trust in automation becomes really low, the IIT becomes significantly large; hence, the real interaction time becomes significantly larger than the DIT. Therefore, the practical free time becomes too small, and the human user is assumed to have too

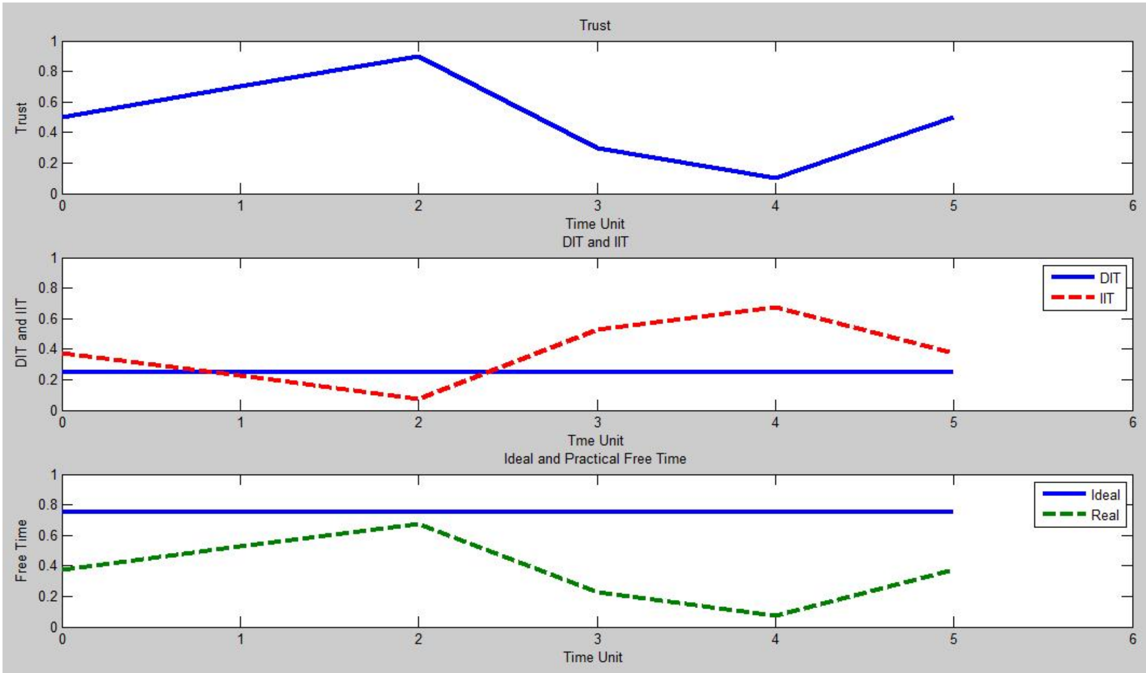


Figure 4.5: Implication of Trust on Interaction time and Practical Free Time

little time to spend interacting with other robots achieving other tasks.

On a related manner, and in order to see the implication of the trust in automation and human reliability factors on the FO metric, Figure 4.6 shows some variations of the human trust in automation with respect to time; the human reliability per time unit is also shown (note that perfect working conditions are assumed, hence the human reliability degradation is only due to the natural human reliability degradation factor as addressed earlier). In this scenario, assuming a DIT of 25%, the ideal FO is 4. However, since FO is no more independent of the human trust in automation and the human reliability, results show that when the trust is high, as in time t = 2, the practical FO becomes closer to its ideal value, while at time t = 4, when the trust is really low, and the corresponding IIT is very high, the practical FO becomes significantly low, and closer to a value of 1, meaning that the human operator has too little time to interact with other robots. It is also worth noting that since both the human trust in automation and the human reliability are time dependent, the FO metric becomes time dependent as well; hence, the system FO is most likely not to take a fixed value throughout the human-robot interaction phase, but to vary with respect to time, depending on the performance

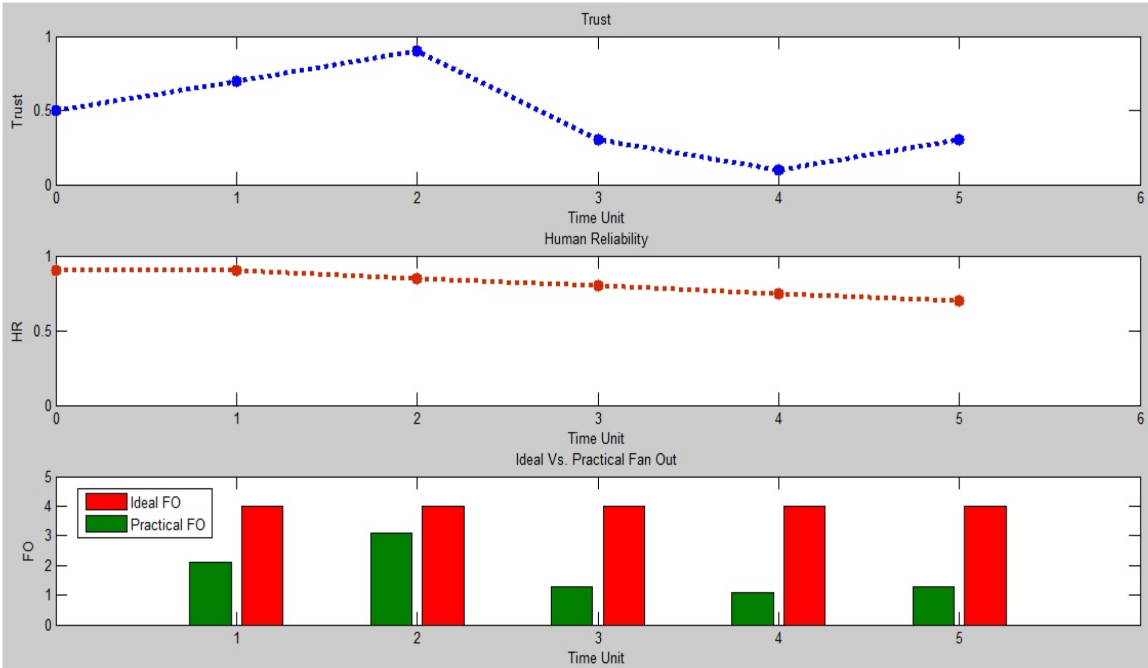


Figure 4.6: FO Calculation

of the human-robot team.

4.2 Multi-Robot System

In this section, we present some simulation results that explain and support our proposed generalized mathematical models for the multi-robot scenarios. Several cases are addressed. Sequential and parallel robot cooperation schemes with varying levels of task dependency are considered.

4.2.1 Sequential Execution of Tasks

In this scenario, two or more robots are cooperating with the guidance of a human user, where only one of the robots is active at a time. The other robots are idle, waiting in turns to be instructed by the human user so they can complete their subtasks. In this scenario, the system FO follows the FO of the active robot. Therefore, the system FO can be defined as the logical OR of the multiple robots' FO, as shown in equation 3.9, where the FO of the idle robot is assumed to take a value of zero. Figure 4.7 shows the system FO for a two-robot system working sequentially toward achieving some tasks. The same result can be extended for other multi-robot systems.

One can also note that the human reliability propagates between active robots while calculating their individual FO, and does not start all over again when the next subtask is carried out by another robot. This is shown in Figure 4.7.

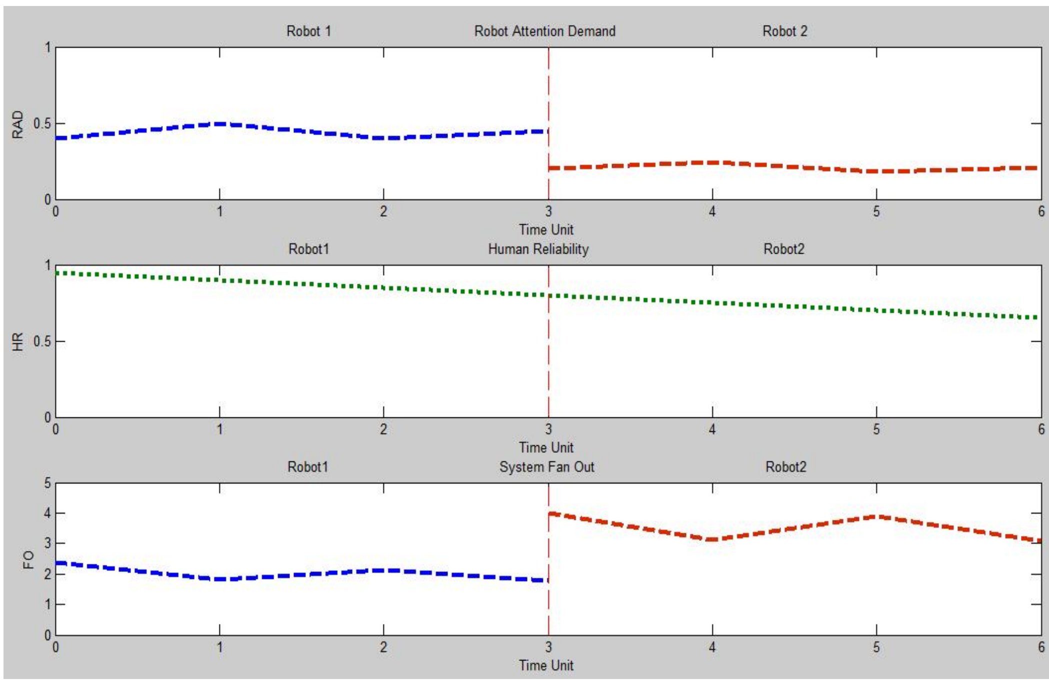


Figure 4.7: Multi-Robot System: Sequential Execution of Tasks

4.2.2 Parallel Execution of Tasks

Independent Execution of Tasks

In this scenario, several robots are cooperating with the guidance of a human user to achieve some tasks. All robots are active simultaneously executing independent tasks, where no task is dependent on another; hence, dependency-related issues can be ignored. In this scenario, since robots might have different contributions toward the overall goal completion, a weighted average method is applied to find the overall system FO, as described in equation 3.10. Figure 4.8 shows the result for a two-robot system, where it is assumed that both robots equally contribute to the final goal completion (50% each).

An important observation to note is that the system complexity is higher than a one-robot system, as the number of subtasks has probably become higher, as well as the mental workload. Therefore, the human reliability is more likely to decrease at a faster rate, as shown in Figure 4.8.

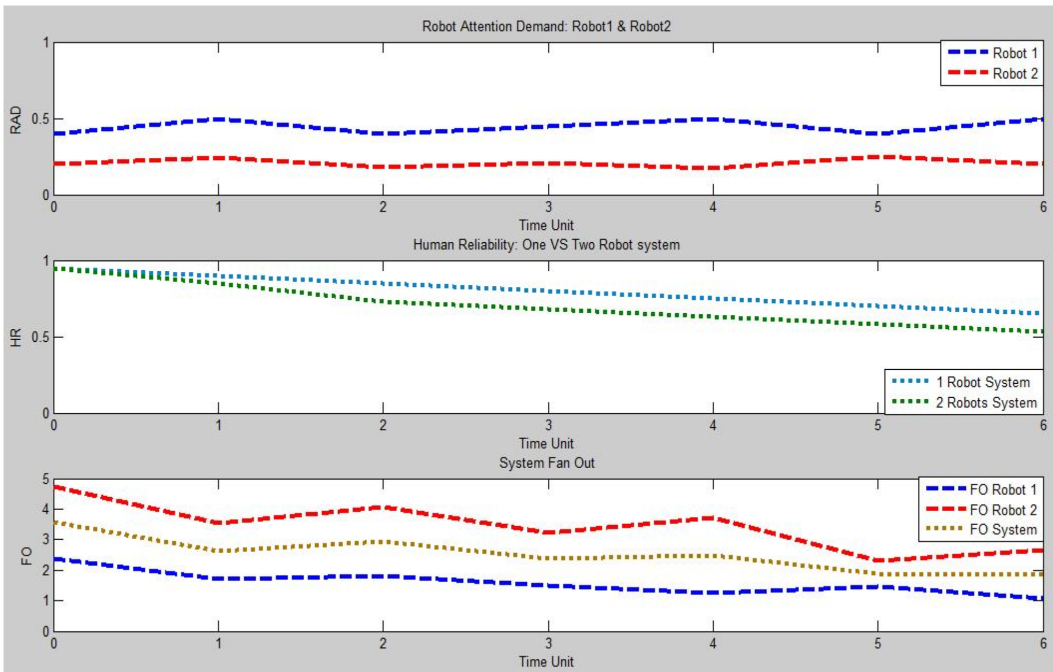


Figure 4.8: Multi-Robot System: Parallel Independent Execution of Tasks

Dependent Execution of Tasks

This scenario is the most complicated compared to the previous ones. Multiple robots are working with the guidance of a human user toward finishing some specific tasks, where all robots are active simultaneously executing dependent tasks. In this case, the system FO is shown to fall somewhere between the weighted average FO and the smallest robot FO. When total (100%) task dependency is encountered, the system FO will be equal to the smallest robot FO, and the less dependency exists, the closer the system FO will be to the weighted average one that corresponds to the task independent scenario. For a two-robot system, the system FO is calculated as shown in equation 3.11. Figure 4.9 shows the result for a two-robot system, where it is assumed that both robots equally contribute to the final goal completion (50% each), and where tasks are 70% dependent. Result shows

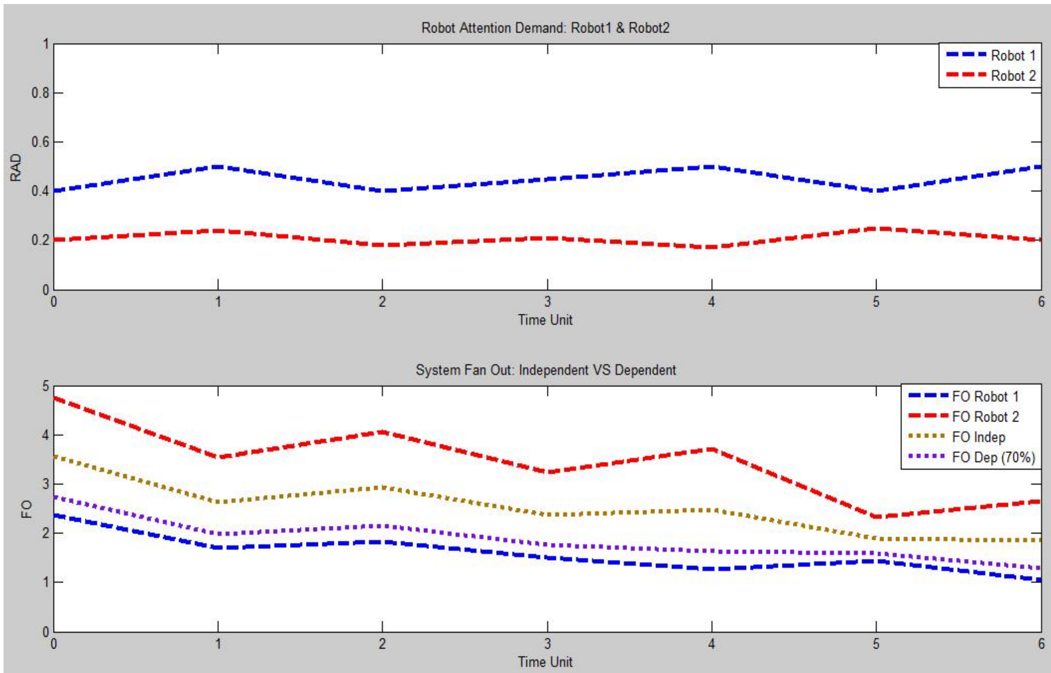


Figure 4.9: Two-Robot System: Parallel Dependent Execution of Tasks

that the system FO falls between the weighted average FO and the smallest FO - closer to the smallest FO though, because percent dependency is higher than 50%.

The system FO is generalized for N-robot systems, as shown in equation 3.12. Figure 4.10 shows a sample simulation result for a three-robot system, where it is assumed that robots 1, 2 and 3 contribute 30%, 50%, and 20% toward the final goal completion. 20% task dependency is assumed between robots 1 and 2, 60% between robots 1 and 3, and 40% between robots 2 and 3. Results show that the system FO falls between the weighted average FO corresponding to the case of task independency, and the smallest robot FO. But since inter-robot task dependencies might highly affect other dependencies in the system - meaning that the dependency between robot 1 and robot 2 might also have some implication and additional cost on the dependency between robot 2 and robot 3, and so on - then the practical system FO is said to be upper bounded by the value calculated in equation 3.12, and lower bounded by the smallest robot FO. This is shown in equation 3.13

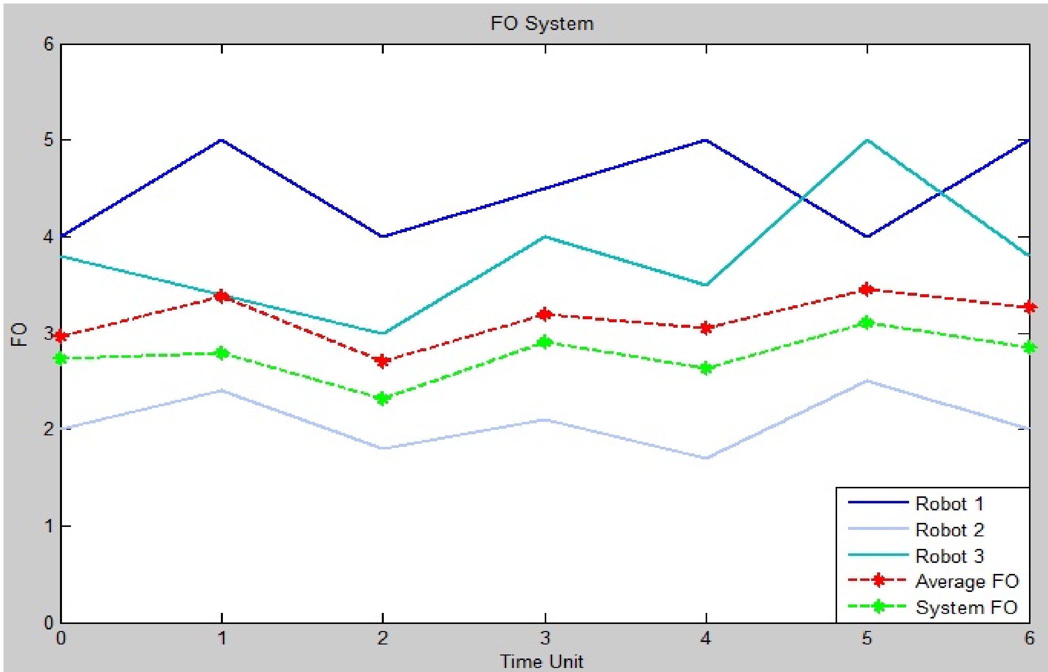


Figure 4.10: Multi-Robot System: Parallel Dependent Execution of Tasks

4.3 Chapter Summary

In this chapter, we discuss some simulation results that intuitively explain and support the importance of our proposed generic metric framework. Two main scenarios are addressed: one-robot and multi-robot systems. Results show the intuitiveness of our proposed fuzzy temporal models that estimate human reliability and human trust in automation. They also present a practical ground that is supported by abstraction and intuition, for the ability of our proposed metric to efficiently assess the performance of both the robot and the human as a team, in a generic way that makes it feasible for this metric to translate well between different applications, as it is not biased toward specific ones. Results also support the three extended mathematical generalization models proposed for the multi-robot systems.

Chapter 5

Real World Environment Setup for Multi-Robot Platform

The knowledge bases presented in this work are based on a human expert's knowledge, and the most recent work in the area of human-machine interaction and performance evaluation metrics. However, and in order to further enhance the proposed system and better represent its knowledge base, an application robotic platform that enables man-machine interaction is implemented, and users' feedback while interacting with the system was noted. The knowledge base is then fine-tuned to better reflect the user's knowledge. For the rest of this work, we further focus on the human trust in automation factor and its implication on RAD and FO correspondingly. The human reliability *shall not be further addressed in this work*, as it requires more in depth studies in the disciplines of sociology, psychology, and physiology, and will be addressed in our future related work.

In this chapter, we discuss the experimental system setup, presenting the main software and hardware components of the robotic platform, along with the proposed design and implementation.

5.1 Software and Hardware Components

5.1.1 PeopleBot

PeopleBot [3], shown in Figure 5.1 is a differential-drive robot that is well known for human-robot interaction projects. It comes with a chest-level extension along

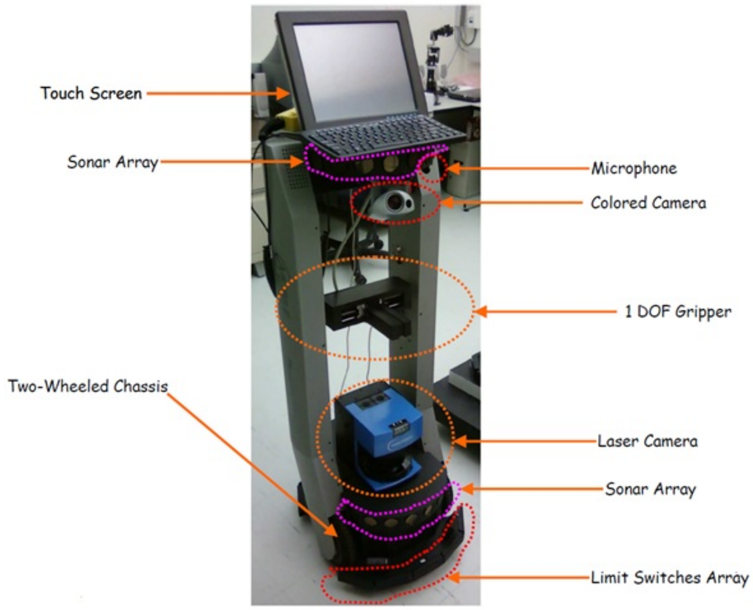


Figure 5.1: PeopleBot Robotic system

with a touchscreen connected to an onboard computer to facilitate interaction with people. PeopleBot is equipped with infrared table sensors and a gripper with sensors which allow the robot to pick up an object from one location and place it at another. PeopleBot features a laser navigation package with an autonomous robotic navigation and localization (ARNL) software that uses Monte Carlo/Markov based techniques for localization and navigation, which allows the PeopleBot to safely navigate autonomously while avoiding obstacles with great precision. PeopleBot can also navigate autonomously, but less accurately, using its built-in sonars. It also comes with a pan/tilt/zoom camera that can be used for object and people recognition, colour tracking, or other robot vision tasks. This can be accomplished using the advanced colour tracking (ACTS) software that comes with the system. Also, with the audio and speech package, it can record and play audio back, and perform speech recognition and speech synthesis.

PeopleBot comes with an advanced robotics interface for applications (ARIA), which offers an API to communicate with all the robot components, control the robot's parameters, and also provides tools to integrate input/output with other custom hardware. It is usable under both Windows and Linux environments, supporting C++/Java/Python programming languages [3]. PeopleBot SDK package also provides some tools for creating maps of the robot's working environment, which is essential for autonomous localization and navigation, or for use by the MobileSim simulator or any software using the ARIA library. C++ was used in our implementation.

Every MobileRobot, including PeopleBot, comes with the Pioneer SDK, which is a collection of libraries and applications, along with other selected accessories. The main components, which will be later described in further details, are [3]:

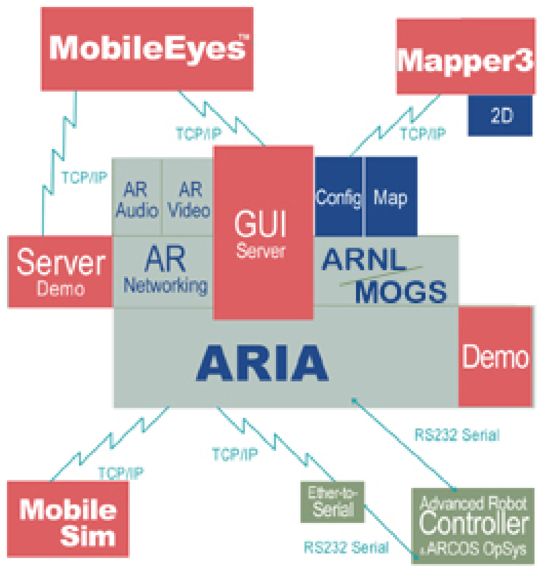


Figure 5.2: PeopleBot Software Components [3]

- ARIA: the advanced robotics interface for applications (ARIA) offers an open source API to communicate with all robot's components, control the robot's parameters, and provides tools to integrate input/output with other custom hardware.
- ARNL: the autonomous robotic navigation and localization (ARNL) adds robot localization and navigation libraries on top of ARIA. ARNL uses Monte Carlo-based techniques for intelligent localization and navigation. This package allows the program to keep track of the robot position, and successfully navigate it to a certain destination.
- MobileSim: MobileSim is a software package whose aim is to simulate MobileRobots platforms along with their environments, which is instrumental for debugging and experimentation purposes using ARIA. MobileSim replaces the

robot's real serial port connection with a simulated control connection accessible via a TCP port using the ArNetworking protocol.

- Mapper3: Mapper3 provides the infrastructure for creating maps of the robot's working environment, which is crucial for intelligent autonomous localization and navigation, and for use in the MobileSim simulator or any software using the ARIA library.

- MobileEyes: MobileEyes is a graphical client software for remotely monitoring and controlling the mobile robot, by connecting to a server program on the robot's onboard computer using the ArNetworking protocol, whose implementation is included with ARIA.

- ACTS: the advanced colour tracking system (ACTS) is a client-server software that processes video frames information to identify and track coloured objects. ACTS can simultaneously track up to 320 independent blobs, over a wide variety of lighting conditions, at the maximum rate of 30 frames per second.

- ARCOS: the advanced robotics control operating system is a low-level software package that manages all the low-level details of the mobile robot's system, such as motor controls, power, firing the sonar, collecting and reporting sonar and wheel encoder data, and other basic processes.

All software is available for both Linux and Windows, and each will be further discussed in the following sections.

Advanced Robot Interface for Applications (ARIA)

MobileRobots' advanced robot interface for applications is an open source C++ library (software development toolkit or SDK) for all MobileRobots platforms. ARIA allows the user to dynamically control the robot's parameters, such as its velocity, pose, relative heading, and other motion parameters through simple low-level commands or through its high-level actions infrastructure. It also provides the proper tools to integrate input/output with other custom hardware and all MobileRobots robot accessories, including the pan/tilt/zoom cameras, pioneer gripper, and more. ARIA also receives all current operating data sent by the robot platform, such as position estimates and laser and sonar sensors readings.

Another library that comes as a part of ARIA is called ArNetworking. This library aims to establish an extensible infrastructure for easy remote network operations of the robots, through user interfaces, and other networked client software. ArNetworking enables clients to connect from any machine on the network to a server executing on the robot's PC, access the robot's data, and issue commands. The ARIA library is written in C++. However, an access to most of the ARIA API is also available from Java and Python languages through wrapper layers. ARIA also includes a variety of other useful tools for building robot applications as separate libraries. These include: speech synthesis and recognition, sound effect playback, mathematical functions, cross-platform (Windows/Linux) thread, and socket implementations [3].

Autonomous Robotic Navigation and Localization (ARNL)

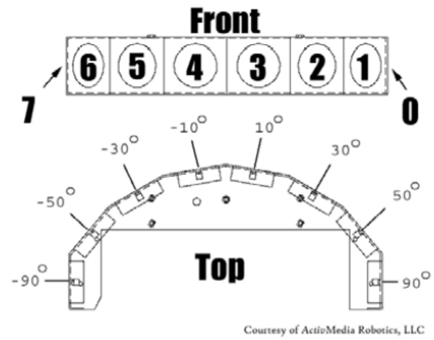
The autonomous robotic navigation and localization is a set of software packages built on top of ARIA, for intelligent localization and navigation. The purpose of this package is to allow the autonomous robot to keep track of where it is, and successfully plans a path to navigate to a certain destination specified either in the control program via the programming interface, or from a remote control client such as MobileEyes via the ArNetworking libraries. ARNL library is also written in C++, and most of the ARIA API is also available from Java and Python languages via wrapper layers. ARNL includes three separate localization techniques implemented as three separate libraries [3]:

- laser localization uses a SICK LMS laser measurement sensor, shown in Figure 5.3(a), to perform precise localization. SICK LMS is an extremely accurate laser distance measurement sensor that is able to provide precise distance readings up to 80 meters, and over a 180-degree area. MobileRobot robots use a Monte-Carlo localization (MCL) algorithm to localize themselves in a map by merging the robot odometry with the laser readings. This is done by the ArLocalizationTask in the ARNL library.

- sonar localization uses the built in sonar sensors, as shown in Figure 5.3(b), for approximate localization. Robots equipped with sonar also localize by merging their odometry with the sonar data based on the MCL. This is done by the ArSonarLocalizationTask in the SONARNL library.

- GPS localization uses a GPS unit to localize the robot within a map. This is mainly for outdoor navigation. The GPS navigation package includes a low profile rugged GPS receiver and LMS laser rangefinder fully installed on the robot. The GPS has an accuracy of 2 meters when used with the standard

(a) SICK LMS



(b) Sonar Array

Figure 5.3: Laser LMS and Sonar Array [3]

free correction technique, but is capable of 1 meter or even 20 centimeter accuracy with the deployment of more advanced correction techniques.

Navigation, on the other hand, is responsible for getting the robot to a precise destination. The path planning module in ARNL is the most important component, whose purpose is to compute a safe path from the robot's current position to a specific destination. Then, the appropriate velocities and heading commands, among other parameters, are sent to the robot so it can follow the computed path as accurately as possible, all while avoiding any unmapped static and dynamic obstacles that could possibly arise in its path. The path planning package uses a grid-based search method to compute the shortest safe path from the present robot location to a specific reachable destination point in the environment map. Even after the main path is planned, ARNL continuously computes an updated version of the planned path so it can plan around any unmapped static and dynamic obstacles it sees within its sensors range. Once the path is computed, ARNL issues the translational and rotational velocities that are necessary to make the robot follow the path as closely and accurately as possible. Obviously, this task would require fairly accurate robot localization, hence the localization task must always be concurrently running with the path planning task to keep track of the robots position.

In addition to the previously mentioned standard path planning and following in a given environment map, the path planning module can incorporate special sectors and actions [3]:

- sectors: sectors are designated areas in the map in which the robot localization and navigation parameters are altered specifically for such areas. For example, one might want the robot to slow down in certain areas and speed up in some other. Several other behaviours are possible. One way corridors are another important example of such areas. These corridors restrict the movement of the robot to only one direction in certain areas of the map. The use of one way areas, however, may cause accessible goals to become inaccessible or be reachable only in circuitous paths.

- re-plan paths: these allow the user to set up the robot to use a method that enables it to re-plan a path when it finds that the current path to the desired destination is blocked. This is useful when rooms can be accessed through multiple doors, some of which may be closed on certain occasions.

- restrictive boundaries: these are basically sectors and lines in the map that cost much more to traverse, and hence the robot has to plan around those areas if possible to avoid such higher cost. These paradigms can be used to force the robot to plan around some objects and areas unless there is no alternative.

- inter-robot communication and mutual avoidance: this service allows multiple robots to communicate with each other about their current status, along with their positions and their projected planned path data, to better avoid each other in the operating multiple-robot environment. Such communication can be either direct in a peer-to-peer fashion, or through a central server.

MobileSim

MobileSim is a software whose aim is to simulate MobileRobots platforms along with their working environments, which is instrumental for debugging and experimentation purposes using ARIA. MobileSim uses line data from a MobileRobots environment map file to simulate walls, obstacles, and other designated areas and sectors in the environment, as shown in Figure 5.4. MobileSim translates a MobileRobots map that is originally created by Mapper3, as we shall describe in later sections, to a stage environment with a simulated robot model. Simulated control connection similar to the real robot's serial port connection is also provided via a TCP port. ARIA, therefore, is able to automatically connect to this TCP port instead of the serial port, making it easy to run and debug the same programs using the simulator before deploying them on the real robot [3].

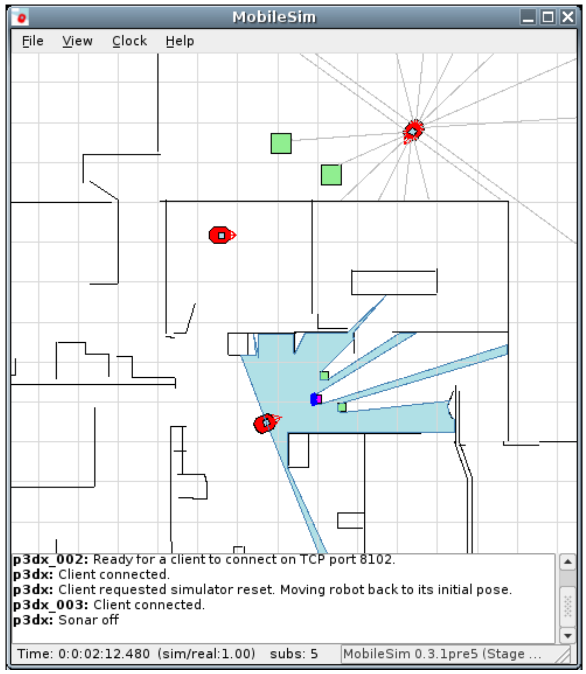


Figure 5.4: MobileSim [3]

Mapper3

Mapper3, shown in Figure 5.5, provides tools for creating maps for the robot's working environment. Such maps can be used for intelligent autonomous localization and navigation purposes, either in the real world by the robot itself, or through simulations using the MobileSim simulator. Toward this goal, the robot has to scan its operating environment using its SICK laser distance measurement sensor. This can be done by manually driving the robot around the environment using a joystick, or remotely using the MobileEyes application. This results in a scan log file (.2d) which can be loaded into the Mapper3 software for further processing. Once the scan is loaded, Mapper3 begins processing the (.2d) file and draws progress into a new map file (.map) that becomes open in Mapper3 for further placing and/or editing of goals, home points, obstacles, entry points for docking stations, and forbidden areas that the navigation software should plan around. [3].

MobileEyes

MobileEyes is a graphical user interface that serves as a client for remotely monitoring and controlling a robot by connecting to a server program running on the

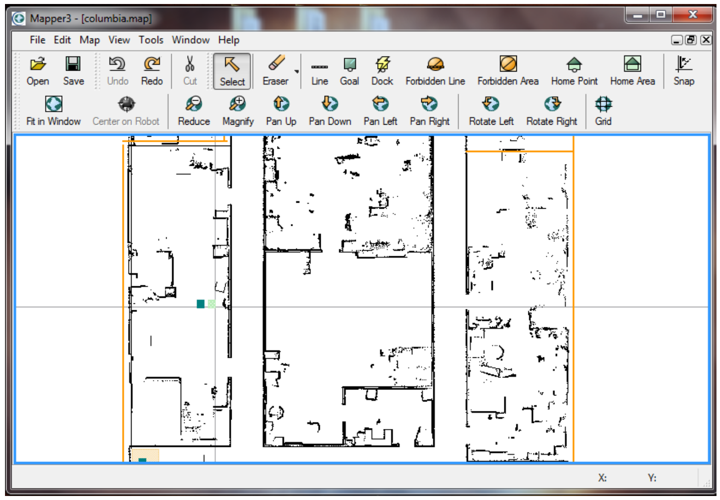


Figure 5.5: Mapper3

robot's onboard computer using the ArNetworking protocol, included with ARIA, or communicating with the simulated robot via MobileSim locally. MobileEyes is able to display the robot's position in an environment map, sonar and laser range sensor data, and other various pieces of information, such as position values and battery voltage. It also displays accessory controls such as camera pan, tilt, and zoom, as shown in Figure 5.6. MobileEyes can also issue commands that control the robot, such as sending it a destination position from the map to navigate to. It also displays the loaded map, and clicking on a destination goal makes the robot plans its path, travels there, and re-plans its path when it detects obstacles. MobileEyes can also change configuration parameters at run time, and connect to multiple servers [3].

Advanced Colour Tracking System (ACTS)

The advanced colour tracking system is a client-server software that processes video frames information to identify and track coloured objects, and send the extracted information to the clients, through an easy to use configure colour-based visual object tracking interface. Equipped with 32 independent channels, ACTS can simultaneously track up to 320 independent blobs, over a wide variety of lighting conditions, at the maximum rate of 30 frames per second [3]. ACTS is very useful as a vision sensor for robotics for object identification and tracking, surveillance, human-robot interaction and many other machine-vision applications. ACTS comes with an integrated training client software, shown in Figure 5.7, that allows the user

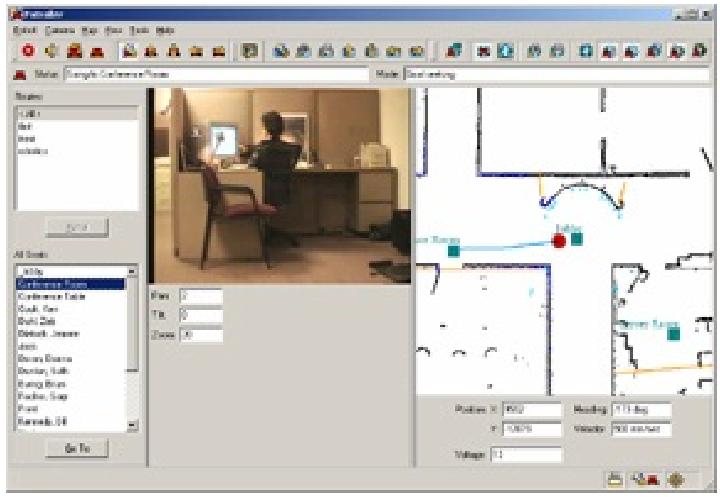


Figure 5.6: MobileEyes [3]

to train ACTS to find and track some coloured objects. The easy-to-use GUI training window allows the user to select and configure the colour tracking parameters, and train the system using either a live image, or a previously saved one.

Advanced Robotics Control Operating System (ARCOS)

The advanced robotics control operating system is low-level software that manages all the low-level details of the mobile robot's system. It handles motor controls, power, operating the motors, firing the sonar, collecting and reporting sonar and wheel encoder data and other basic processes, all on command from and reporting to a separate client application via ARIA. ARCOS is also responsible for monitoring and responding to protection and emergency triggers; for example, ARCOS server initiates a stall in the robot when one or more bumper segments get triggered during the movement of the robot in some direction. ARCOS also contains a protocol that will halt the robot's motion if the communication between the client and the servers running on the robot's onboard computer is disrupted for some interval of time [3].

5.1.2 Verbal Interaction

Vestec's automatic speech recognition engine (VASRE) [139] was used in this work to enable human-robot interaction via semi-natural speech. VASRE is a speaker-independent speech recognition engine that supports a distributed architecture of

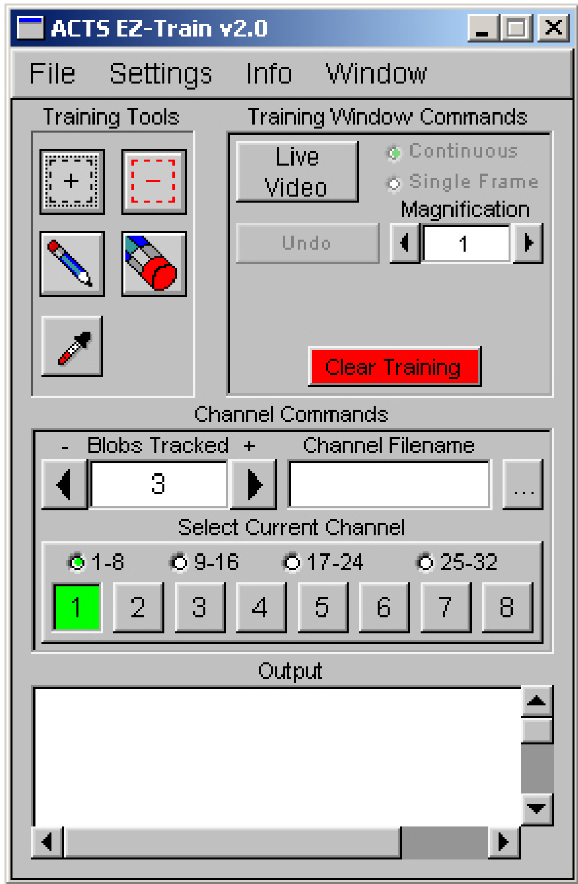


Figure 5.7: ACTS

servers and clients. VASRE supports multiple languages, large vocabulary, and continuous speech recognition. Its acoustic models were trained based on continuous hidden Markov modelling. Equipped with noise reduction techniques and voice detection algorithms, it ensures smooth data input and more accurate speech recognition. Its simple-to-use API allows easy integration with application programs. The output of the engine contains such information as the raw recognized text, confidence scores, and logical parsing for generating semantic results.

The main components of the VASRE speech engine are: recognition servers, resource managers (RM), and grammar compilers.

Recognition Servers

Equipped with acoustic models trained based on continuous hidden Markov modelling, the recognition server of VASRE supports multilingual, speaker-independent, large vocabulary, and continuous speech recognition acoustic models. The recognition server accepts speech recognition grammars in two formats: a text grammar and a binary grammar. A text grammar includes the speech grammar rules written in an Augmented Backus-Naur Form (ABNF) or an Extensible Markup Language (XML) format. Compiling a text grammar file into a binary file can be done using the grammar compiler. VASRE is able to process either batch audio or streamed audio. VASRE is equipped with noise reduction and voice detection algorithms to ensure smooth data input and thus better recognition results. The communication session between a server and a client is called a port. The client initiates its communication process with the server by opening a port. The server originally has no speech grammars; therefore, the client is responsible for loading the server with the appropriate grammar files. Then, audio files can be passed to the server for recognition. The VASRE server can only process 8 kHz linear PCM audio. The output of the engine contains such information as the raw recognized text, confidence scores, and logical parsing for generating semantic results. A VASRE system is able to support multiple servers, where each server can support a single client at a time [139].

Resource Manager

The resource manager of VASRE is the central monitoring and management unit that manages speech recognition sessions and coordinates communications between recognition servers and clients. At system start-up, the RM reads the number of servers and the grammar size, and monitors the status of the machines in the local network and recognition servers running thereon. The RM performs load balancing by assigning a given client request to an idle server on the least busy machine.

Grammar Compiler

A VASRE speech recognition server accepts both text and binary grammar files. The grammar compiler is responsible for converting an easy-to-read text grammar into the binary format, which can be directly loaded into the recognition server. The text grammar describes the words and sentences that the engine should recognize from the given audio. Both text and binary grammars can be added to the server at run time, however, it is a better strategy to add binary ones, as this releases the server from doing extra work and possibly delaying performance.

The grammar compiler generates the pronunciations of words used in the text grammar by referring to a predefined lookup table that covers most common English terms; for those undefined in the table, the user is allowed to either rely on an auto pronunciation engine, which guesses the pronunciations of given words, or manually specify the pronunciations using predefined phonetic symbols. Each grammar added to the server has two fields: tag, and activation token. The server can load and use multiple grammars at a time, and all active grammars are used for speech recognition. The activeness of each grammar is controlled by the client. Based on active grammars, the server processes the audio file to find the best match. All grammars are automatically deleted when the client program closes the port [139].

5.2 Platform Design and Implementation

The application robotic platform supports a distributed architecture for reliable and scalable operation of clients and robot servers. The proposed distributed architecture comprises three components: the robot server, client, and the resource manager (RM). The server and RM are the permanent components of the distributed architecture. The RM is the control tower of the distributed architecture. It manages one or more robot servers and coordinates communication sessions between servers and clients. The robot server has two states: busy or idle. A server is busy if it is executing an action upon receiving a client command. A server is idle if it is not busy. Under the idle state, the robot server periodically communicates with the RM to report its status. The RM balances server loads over different machines. For example, if a robot server A is loaded with several queued commands, the RM guides the next client request to a robot server B that has similar capabilities, if available. The recognition client initiates a communication session with the robot server by asking the RM about the server's availability and functionality.

5.2.1 Resource Manager

The resource manager is the central entity of the framework that connects the various components to one another. As each of the robotic entities turn on, it contacts the RM registering itself along with its capabilities. This is shown in Figure 5.8.

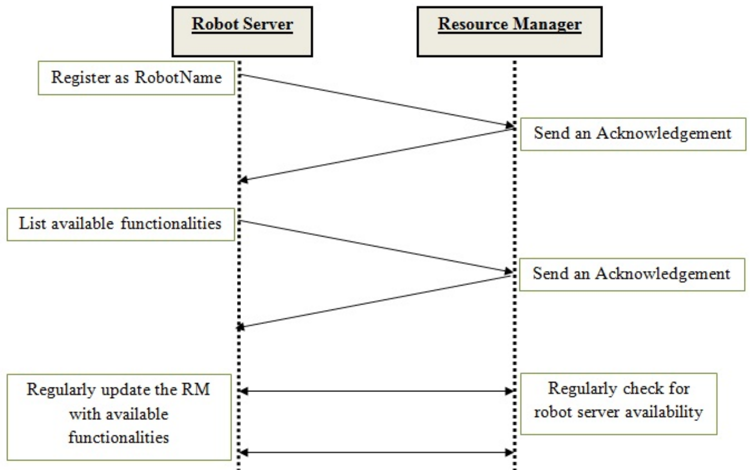


Figure 5.8: Robot Server Communication with Resource Manager

In this way, the RM keeps a record of all active robots and their states, and this information can then become available to other clients and interfaces by request.

The RM is also the main entity that communicates with the client. The recognition client initiates a communication session with the RM. It contacts the RM registering itself and receiving a client ID. In this way, the RM keeps a record of all active clients. Then after a command is ready on the client side, the client contacts the RM asking for a command dispatch. The client might specifically ask for a

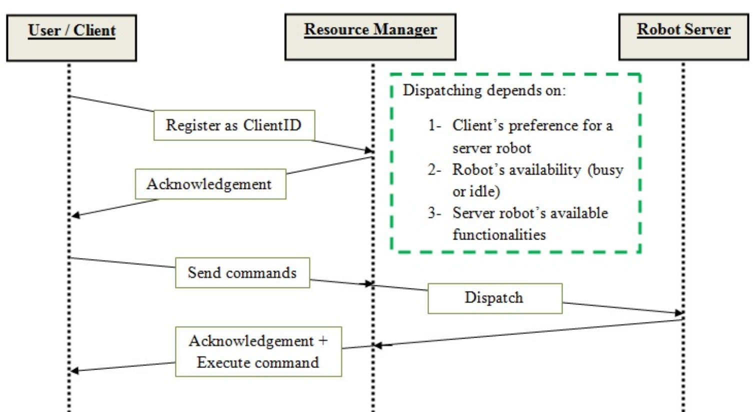


Figure 5.9: Client Communication with Resource Manager

specific robot, or might send a generic command that is to be executed by the first available server robot with such ability to perform the task. This is illustrated in Figure 5.9.

The RM keeps a queue of instructions for each robot. This allows multiple robots to be controlled simultaneously. Figure 5.10 illustrates a simplified data flow view of the RM. The robot store contains records of all currently connected robots, their actions, locations, along with various other information. By keeping a record of this on the RM rather than retrieving it every time it is requested, some amount of network traffic is eliminated. The task queue contains sequences of tasks that need to be executed. It consists of a list of parallel tasks, where each parallel task is a collection of tasks that need to be executed in series. The scheduler, on the other

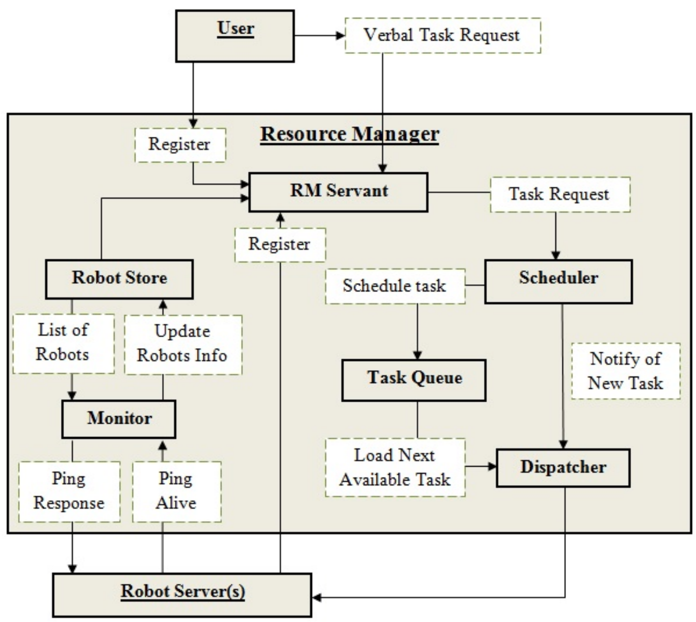


Figure 5.10: Generic Resource Manager Design

hand, is responsible for receiving task requests from clients and loading them into the task queue in a specific order. The dispatcher loads tasks out of the task queue and sends them to individual robots as they become available. The dispatcher can handle numerous parallel tasks simultaneously. Finally, the monitor is a separate thread that runs in the background and periodically checks with every robot to make sure it is still connected and if any information has been updated. If a change has happened, the robot store will be updated on the RM.

5.3 Chapter Summary

The knowledge bases presented in this work are based on a human expert's knowledge, and the most recent work in the area of human-machine interaction and performance evaluation metrics. In this chapter however, and in order to further enhance the proposed system and better represent its knowledge base, an application robotic platform - that enables man-machine interaction via semi-natural language to complete tasks with varying levels of complexity - is implemented. User feedback is recorded and used to tune the knowledge base where needed. The main hardware component used for such purpose is the PeopleBot robot. Verbal communication with the robot is enabled through the use of Vestec's automatic speech recognition engine (VASRE).

The application robotic platform supports a distributed architecture for reliable and scalable operation of clients and robot servers. The proposed distributed architecture comprises three components: the robot server, client, and the resource manager (RM). The RM is the control tower of the distributed architecture. It manages one or more robot servers and coordinates communication sessions between servers and clients. The RM also balances server loads over different machines. For example, if a robot server A is loaded with several queued commands, the RM guides the next client request to robot a server B that has similar capabilities, if available.

Chapter 6

Experimental Results and Metric Assessment

In this chapter, we further focus on the two-level human trust in automation factor, along with its proposed knowledge bases. The human reliability factor is not further addressed throughout this chapter as it requires more in depth studies in other disciplines that should be carefully considered, which goes beyond the purpose of this work, and hence noted to be addressed in other future related work.

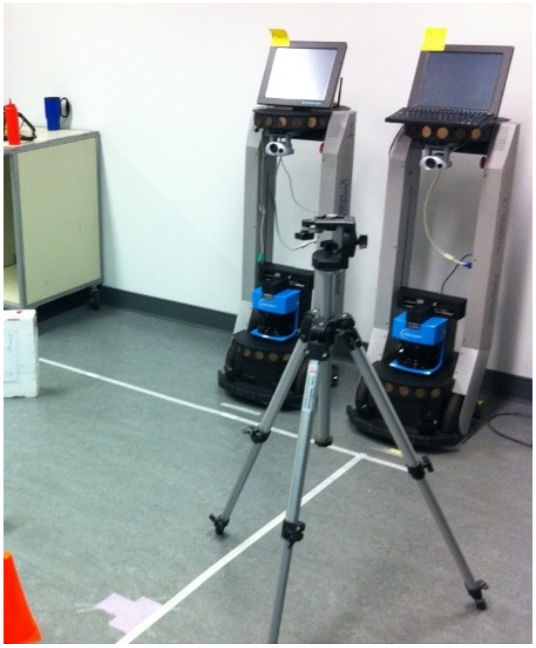
6.1 Suggested Set of Experiments

The set of experiments conducted in this work involves two Peoplebot robots, working singly or together toward achieving some tasks, and a human operator. The purpose of these experiments is to support the correctness and the validity of the proposed fuzzy knowledge base, and tune rules where needed to best accommodate and represent the human expert's knowledge. A sample of nine users was chosen for this purpose, and each was exposed to a set of five to six scenarios where the robot attempts to complete a set of different tasks, with varying levels of success, under the command and operation of the human user. Human trust in automation, along with other first- and second-order perceptions, are marked at different time units and compared to those obtained/inferred using our proposed framework. The scenarios emphasize how the human trust in automation varies with time, according to the system's success fulfilling the required tasks. The tasks vary from simple to more complex. In some scenarios, the robot is instructed to perform a series of simple tasks of moving a certain distance forward or backward, turning left or right at a certain angle, and/or controlling its gripper. The robot is instructed via the user's natural speech. More complex tasks require the robot to pick an object from a certain location and place it at a goal location. In doing so, the robot has to navigate the environment, avoiding static and dynamic obstacles, looking for the object. Once it is located, the robot gets near the object, performs a series of gripper actions (moves the gripper to an appropriate height, opens the gripper, and finally embraces the object) and navigates to the goal destination to place the object at the desired location. This is illustrated in Figure 6.1. Other tasks require the robot to build a map for its environment. In doing so, the robot has to wander in the environment, gathering all sonar and laser sensor-based measurements, detecting and avoiding obstacles, and then finally converting the gathered data into a two dimensional map for the working environment. Another scenario could require the robot to locate, grab, or follow a predefined coloured object in a room. Further details on the experimental setup for the user feedback assessment can be found in appendix D.

6.2 Domain-Specific Measures

Users' observation of the first-order perceptions depends on domain-specific measures. For example, for any mobile system, the ability of the robotic agent to navigate in its working environment is one of the most crucial capabilities of all. Staying operational, avoiding dangerous situations such as collisions with static and dynamic obstacles, and staying within safe operating conditions come first. Navigation is a fundamental task for mobile robots: move the robot from A to B. Performing this task requires determining where the robot is (A), where it needs to be (B), how it should get there (path planning and resource usage), and how to deal with static and dynamic environmental factors and contingencies (obstacles and hazards) encountered on the way. And of course, the objective of this task is to perform work that requires significant "social interaction", where both the human and the robot collaborate to accomplish the desired task.

Not all navigation though is without problems. Obstacles are often encountered on the projected planned path, and at times, robotic systems may find themselves stuck in ditches or debris, and hence have to extract themselves from such situations. Creating a plan for extraction requires the system knowing the characteristics of the obstacle (size, hardness) as well as knowing other potential hazards in the surrounding environment. This process involves interpretation of sensor data, detection and identification of objects and obstacles, judgment of sizes and distances,



(a) Ready to Go

(b) Object Detected and Grabbed

(c) Navigating to Destination

(d) Object Dropped at Destination

Figure 6.1: Pick and Place

and judgment of motion. Some potential measures of this factor include [10]:

- detection measures.
- recognition measures.
- classification accuracy.
- relative and absolute judgments of distance, size, or length.
- platform relative judgments - how long would it take the robot to reach destination B?
- estimates involving relative motion - will the robot perceive other moving objects?
- obstacles that were successfully avoided.
- obstacles that were not avoided but could be overcome.
- identification errors - number of incorrect targets or number of targets missed.
- estimates involving the ability of the robot to recover from faults.
- the ability of the robot to realize that it reached the wrong destination, thus deploying a recovery strategy.

These potential measures are domain-specific, and they represent the tool to estimate the first-order perceptions related to fault size: fault frequency (e.g. how often does the robot hit an obstacle, misidentify objects, reach a wrong destination, miss a target, stall and ask for the operator's help?); fault cruciality (e.g. did the robot collide with another robot or a human being, or just missed an object or identified a wrong target? did the robot lose its way and leave the designated area to a road full of traffic or did it just stall and ask for the operator's help?); and fault recovery (e.g. after reaching a wrong destination because of deviation from trajectory, will it be able to recover its reference position?).

System Awareness, on the other hand, is another equally important issue. The system needs to have an overall understanding of the locale in which it is working. Some parameters might need to be tuned prior to the start of the task or mission, as in whether the robot is operating in an indoor or outdoor environment, off the road or on the road, in an urban terrain, wooded terrain, or desert. During task execution, the system needs to know where it is, and the nature of the particular area it will be operating in. For example, if the robot is navigating inside a building, the system should know on which floor it is located. Further, the mobile navigator must have the up-to-date knowledge about its current abilities and limitations, as the less a robot is aware of its capabilities and the less it is able to recognize when it is having trouble, the more human monitoring and intervention is required (enter closed-loop paradigm). Finally, a good understanding of the human operator's presence, status, location, and abilities is a must as well.

Some potential measures of this factor include:

- the robot's understanding of how their tasks should be completed.
- the robot's awareness of who is communicating with whom.
- the robot's knowledge of such things as other agents' roles and responsibilities, positions and status, and capabilities and limitations.
- the robot's awareness of what the human operator knows and what they are doing.
- the robot's ability to perceive the elements in their working environments, comprehend their meaning, and project their status in the near future.
- the robot's amount of information about the presence and activities of people and machines in a shared environment.
- the robot's awareness of simultaneous activities performed to achieve a shared task.
- the robot's up-to-minute knowledge of other participants' interactions with the shared workspace.
- the robot's understanding of its own capabilities and limitations, and when it is a good time to stop and ask for the human operator's help and guidance.

These potential measures, among others, represent the tool to estimate the first-order perceptions related to awareness: machine awareness of its capabilities (e.g. will the mobile robot attempt to fly if it were asked to? will it know when it is stuck and should ask for the operator's intervention?, as shown in Figure 6.2); context awareness of the task (e.g. does the mobile robot know what the task is about, and how it can be completed?); and machine awareness of the human operator's

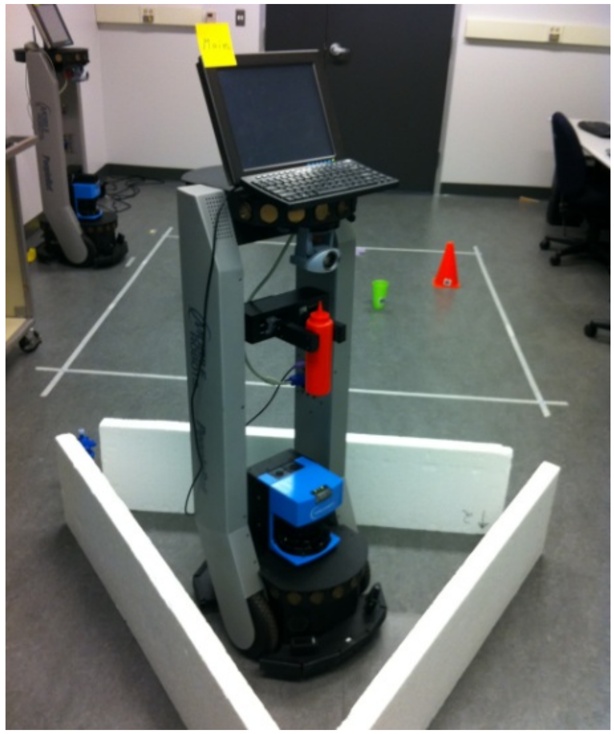


Figure 6.2: Cul-De-Sac Scenario: Dead End with No Way out but a Human Intervention

availability and cognitive and physical abilities and limitations, which we refer to as the human awareness (e.g. does the robot know where the human operator is? what they are doing? their roles and responsibilities, and if they are available or able to help out?)

Finally, as for productivity, measures that refer to the system effectiveness and how well the task is completed, as well as its utility, benefit, and/or importance, are involved.

Some potential measures include:

- percentage of navigation.
- tasks successfully completed.
- coverage of area.
- deviation from the planned route.

- overall percentage of detected true targets.
- time to complete the task compared to the operator's time and effort to complete the same task individually - ratio of operator time to robot time.
- safety measures during task completion - any critical losses and damages?
- utility of the achieved task and its implication on other related tasks.
- number of unplanned operator interventions per unit time.

These potential measures, among others, will represent the tool to estimate the first-order perceptions related to productivity: task completion (e.g. percentage of navigation tasks successfully completed, coverage area); and task sophistication and utility (e.g. ratio of operator time to robot time, number of unplanned operator interventions per unit time, implication of successful/unsuccessful task completion on the overall goal).

6.3 Users Feedback vs. Knowledge Base

Users' perceptions are helpful to enrich the expert's knowledge base, make sure it reflects a representative knowledge, and provides feedback on scenarios that could have been given lower attention at implementation time. Several rules were tuned after receiving feedback from users. Some rules belonging to the productivity knowl-

Table 6.1: Productivity % Error Reduction

Subject	Old Rules	New Rules	% Error Reduction
Subject#1	2.28	2.28	0.00
Subject#2	11.40	9.27	2.13
Subject#3	5.05	3.90	1.15
Subject#4	6.03	6.03	0.00
Subject#5	9.92	5.36	4.56
Subject#6	11.59	4.45	7.14
Subject#7	5.02	5.02	0.00
Subject#8	8.14	8.14	0.00
Subject#9	9.40	9.67	-0.27

edge base were tuned when feedback showed that users tend to give more weight

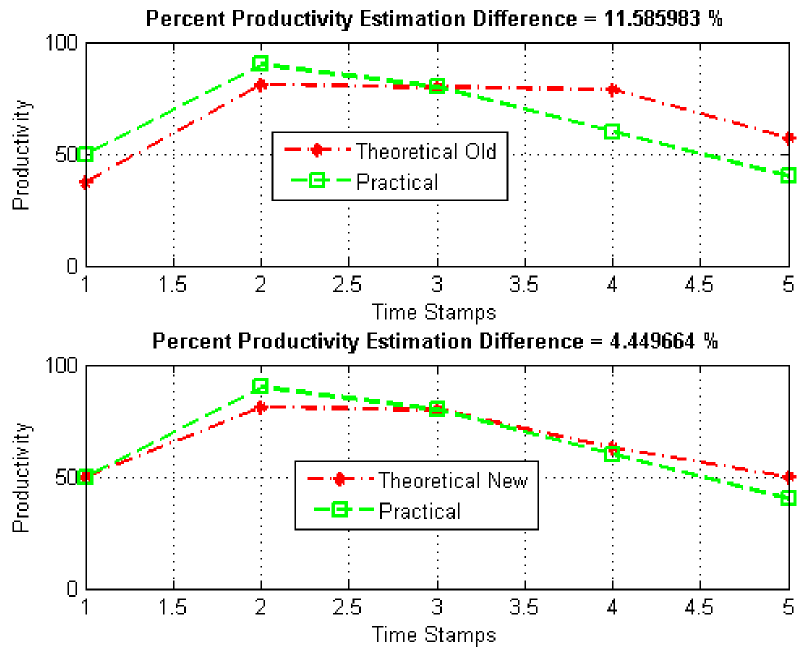


Figure 6.3: Percent Productivity Estimation Error: Before and After Rules Tuning (Sample User)

toward task completion than task complexity and sophistication. Figure 6.3 shows a sample simulation for one user, and the implication of this rule tuning on the overall percent estimation error between the practical feedback (received from the user) and those inferred using the Mamdani fuzzy inference model for the productivity factor. Table 6.1 shows the implication of such tuning on the remaining sample users. The results show some overall significant reduction of error when tuned rules are put in place.

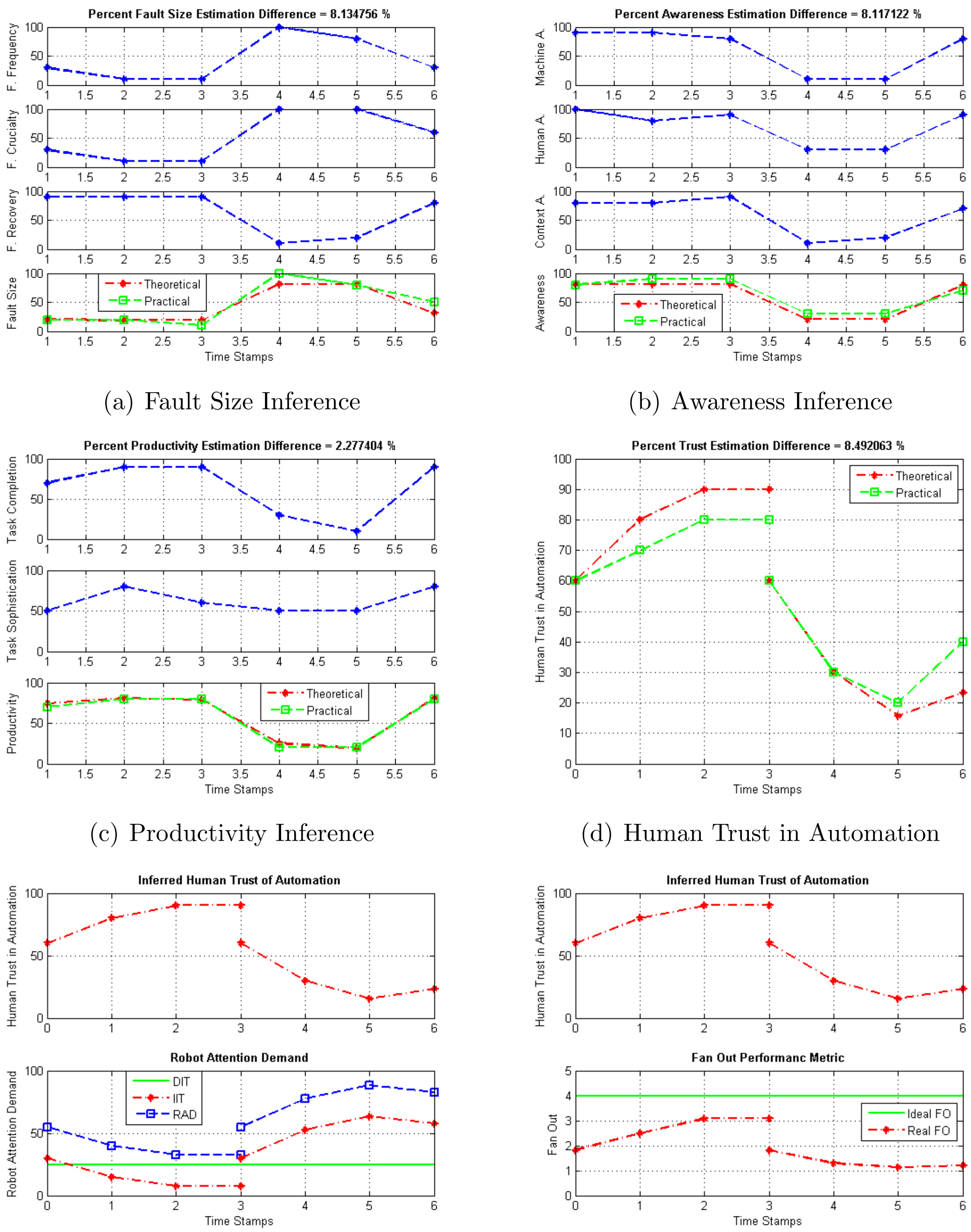
Similar findings were also reported for the trust inference mechanism at level II of the proposed framework. User feedback showed that users tend to generally build trust rather more slowly than earning it, but when the trust is already at a *very low* state, this build up process becomes a bit slower yet. Therefore, a few rules belonging to the knowledge base representing the state *very low* at level II were altered to further accommodate such observation. The implication of such tuning is reported in table 6.2. Results also show some overall significant gain in the approximation accuracy.

Table 6.2: Trust % Error Reduction

% Subject	Old Rules	New Rules	% Error Reduction
Subject #1	6.43	8.49	-2.06
Subject #2	6.38	3.04	3.34
Subject #3	7.83	4.16	3.67
Subject #4	5.33	2.00	3.33
Subject #5	6.40	6.40	0.00
Subject #6	10.60	10.60	0.00
Subject #7	8.86	8.86	0.00
Subject #8	5.29	5.29	0.00
Subject #9	15.14	9.14	6.00

Figures 6.7(a), 6.7(b), 6.7(c), and 6.7(d), for instance, show a comparison between the theoretical results obtained using our proposed two-level trust evaluation framework, and the practical ones obtained from one sample user. Two independent sets of three scenarios each took place. The sets are independent and separate which explains the discontinuity at time stamp $t = 3$, which represents time stamp $t = 0$ for the second set. The first set focused on good robot performance, and successful task completion. The user's trust evolution was noted. In the second set, the user is asked to start interacting (starting with the same initial human trust in automation at time $t = 0$) with the robot with a different set of scenarios, which focused mostly on poor robot performance, followed by an instant significantly improved task execution. The user's trust in the system automation was also noted. Figure 6.7(a) shows the user's first-order perceptions of fault frequency, fault cruciality, and fault recovery, along with the overall fault size. The latter value is compared to that obtained using our fault size fuzzy inference model. Figures 6.7(b) and 6.7(c) address the same matter for both the awareness and the productivity factors. Figure 6.7(d) compares the trust value as noted from the user and generated using our proposed fuzzy level II. Results show accurate trust approximation and good inferences in both levels I and II, which reflects proper and representative knowledge bases design.

Figure 6.7(e) shows the implication of the human trust in automation factor on both indirect interaction time (IIT) and robot attention demand (RAD). Direct interaction time (DIT) is assumed to be 25% of the overall task time, during which the human user is to instruct and inform the robot about the task to be completed. Results show that when the human trust in automation increases, the indirect



(a) Fault Size Inference

(b) Awareness Inference

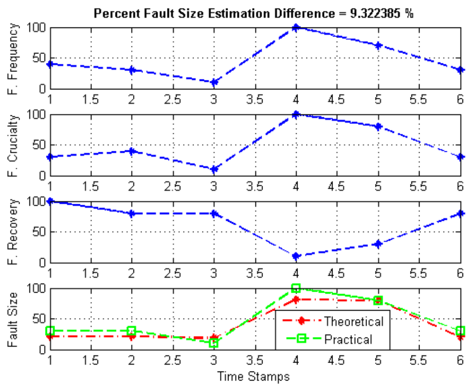
(c) Productivity Inference

(d) Human Trust in Automation

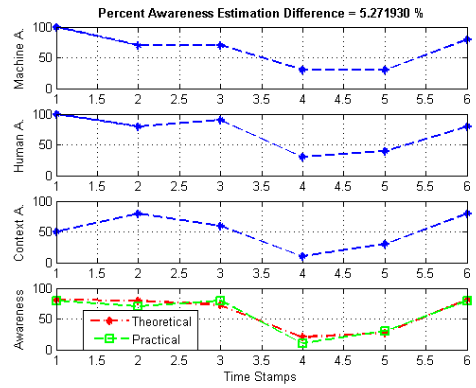
(e) Robot Attention Demand

(f) Fan-out

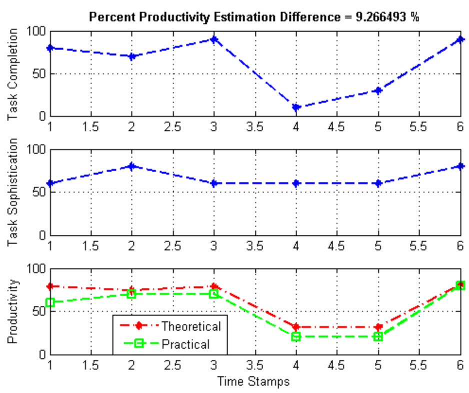
Figure 6.4: Subject #1 - Levels I and II Inferences



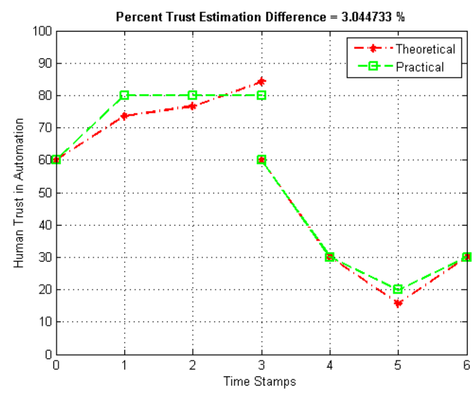
(a) Fault Size Inference



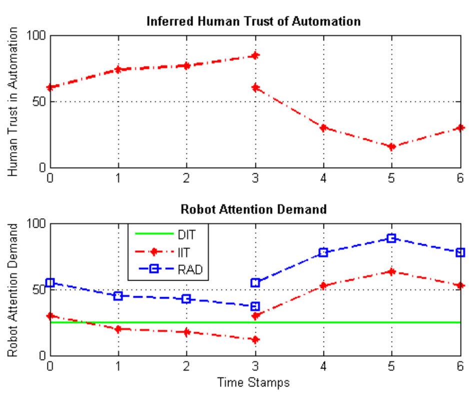
(b) Awareness Inference



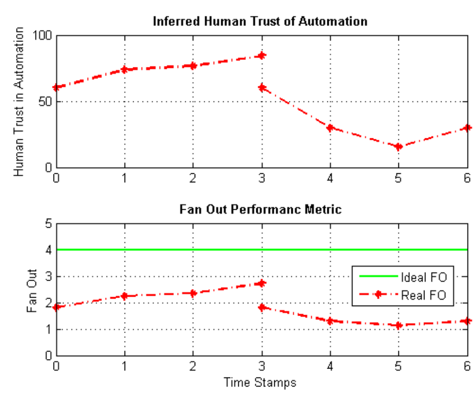
(c) Productivity Inference



(d) Human Trust in Automation

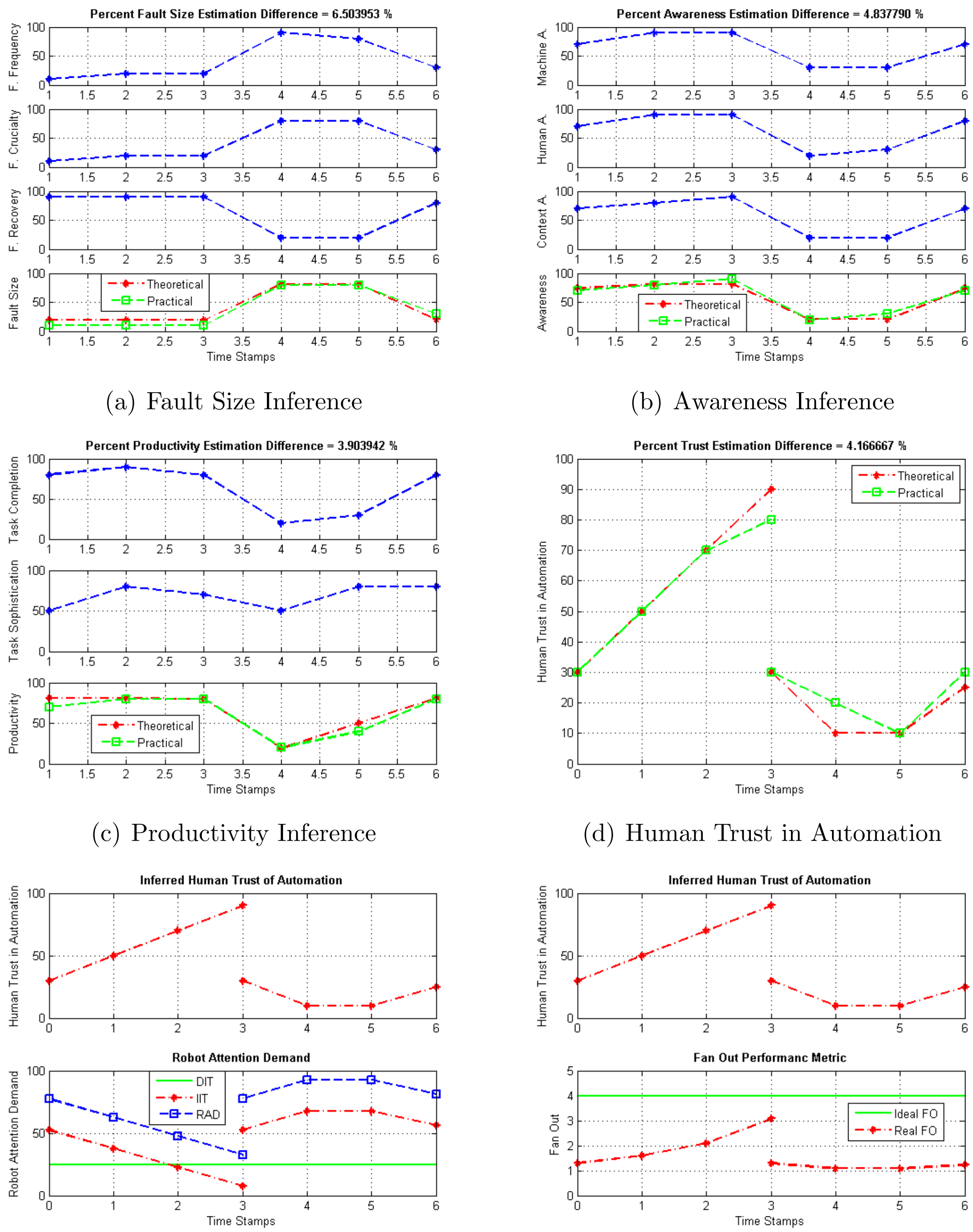


(e) Robot Attention Demand



(f) Fan-out

Figure 6.5: Subject #2 - Levels I and II Inferences



(a) Fault Size Inference

(b) Awareness Inference

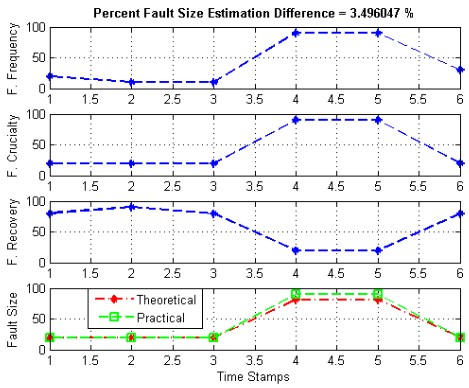
(c) Productivity Inference

(d) Human Trust in Automation

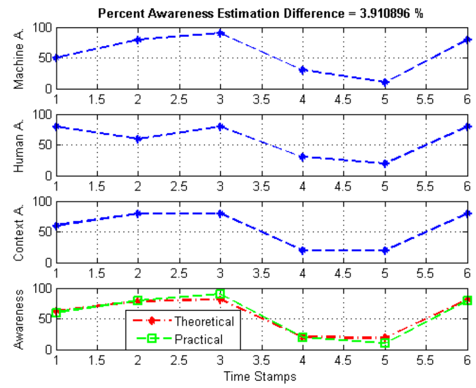
(e) Robot Attention Demand

(f) Fan-out

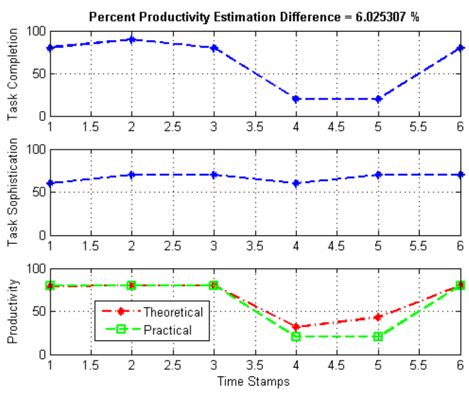
Figure 6.6: Subject #3 - Levels I and II Inferences



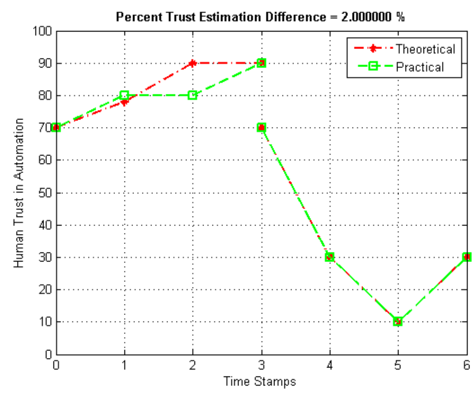
(a) Fault Size Inference



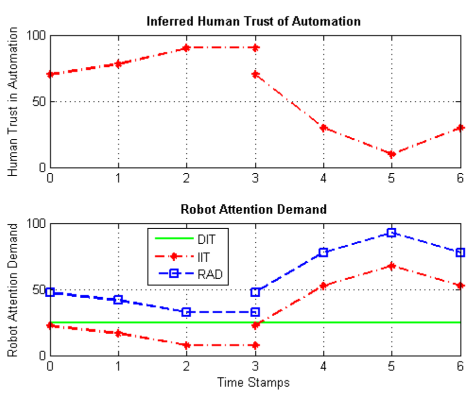
(b) Awareness Inference



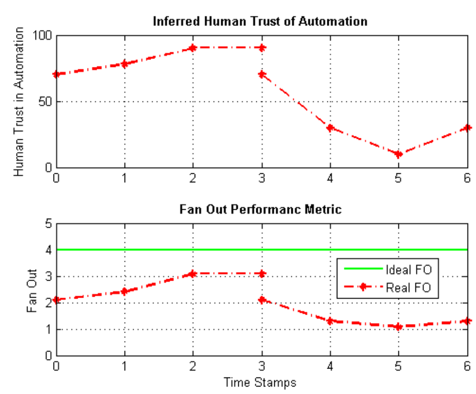
(c) Productivity Inference



(d) Human Trust in Automation



(e) Robot Attention Demand



(f) Fan-out

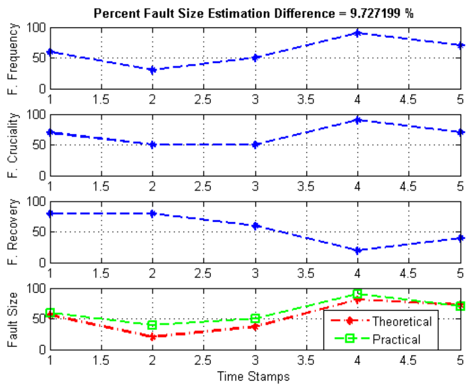
Figure 6.7: Subject #4 - Levels I and II Inferences

interaction time (IIT) spent monitoring the robot and interfering when needed, decreases, and vice versa. Figure 6.7(f) shows the same variations of the human trust in automation with respect to time, along with its corresponding FO metric value. In this scenario (assuming a DIT of 25%), the ideal FO is 4. However, since FO is not independent of the human trust in automation, when the trust is high, the practical FO is close to its ideal value, while when trust is low and the corresponding IIT is very high, FO is far below its ideal value, and closer to 1; thus, the human user is assumed to have too little time to interact with other robots. The same reasoning can be applied to Figures 6.4, 6.5, and 6.6.

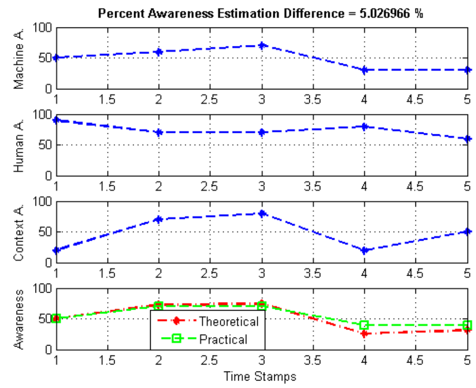
Similar results are also shown in Figures 6.8, 6.9, 6.10, 6.11, and 6.12, where five randomly selected continuous scenarios took place with varying levels of success and completion. User feedback was noted and compared to the inferred values. Results also show that our proposed system, with its set of modified rules, is representative and within reasonable accuracy. Table 6.3 shows the results for all nine sample users selected in this work. The table shows the approximation errors for all inferred values, starting from fault size, and including awareness, productivity, and finally human trust in automation. The results are very encouraging for the correctness of the knowledge base. Future work will include more users taking part in this work, providing feedback that further better represents the proposed knowledge bases, and interacting with different types of other robots.

Table 6.3: Inference % Approximation Errors

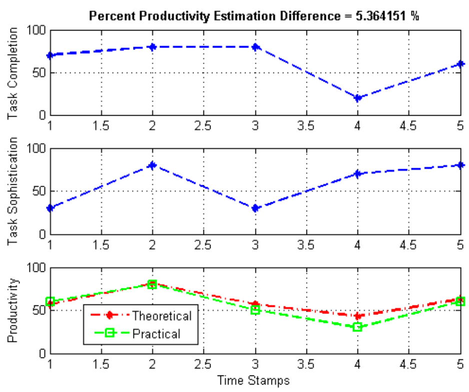
% Error	Fault Size	Awareness	Productivity	Trust
Subject#1	8.13	8.12	2.28	8.49
Subject#2	9.32	5.27	9.27	3.04
Subject#3	6.50	4.83	3.90	4.16
Subject#4	3.50	3.91	6.03	2.00
Subject#5	9.73	5.03	5.36	6.40
Subject#6	4.61	8.90	4.45	10.60
Subject#7	10.72	9.84	5.02	8.86
Subject#8	7.48	7.62	8.14	5.29
Subject#9	9.18	7.31	9.67	9.14
Avg Error	7.69	6.76	6.01	6.44
Std Dev	2.28	1.94	2.37	2.84



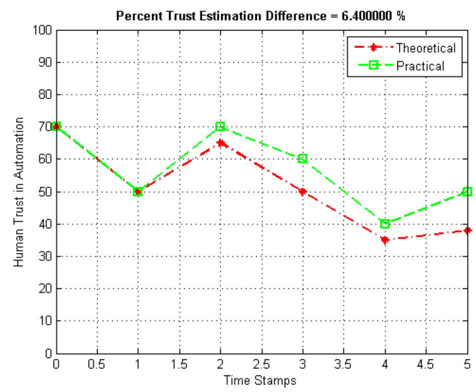
(a) Fault Size Inference



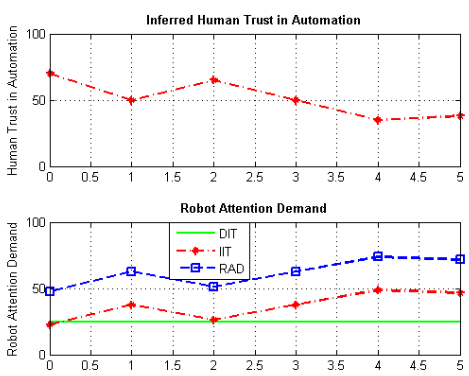
(b) Awareness Inference



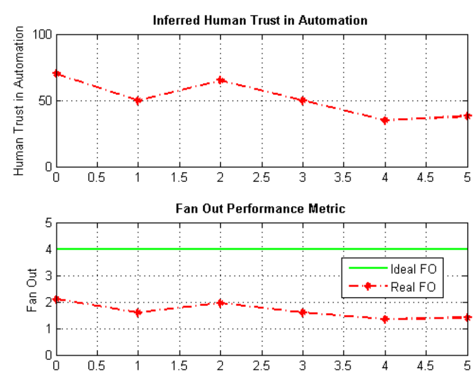
(c) Productivity Inference



(d) Human Trust in Automation

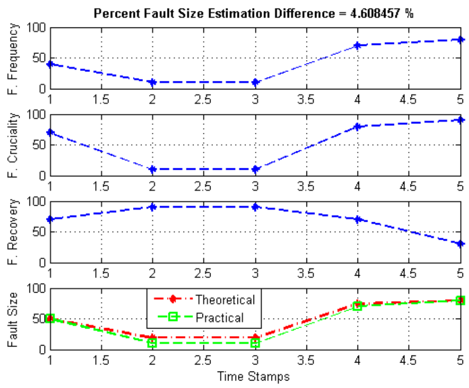


(e) Robot Attention Demand

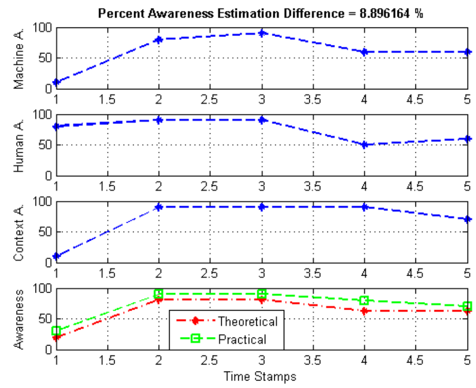


(f) Fan-out

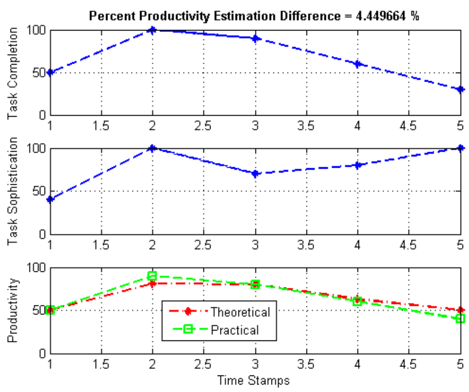
Figure 6.8: Subject #5 - Levels I and II Inferences



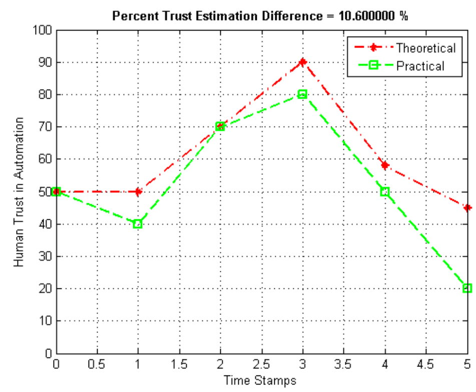
(a) Fault Size Inference



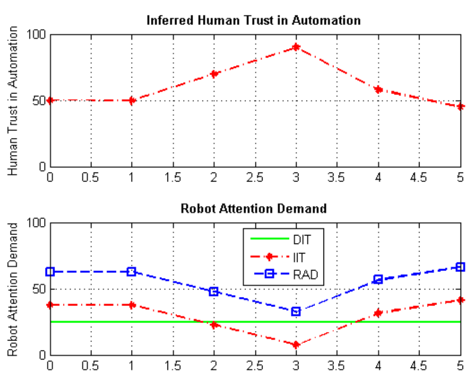
(b) Awareness Inference



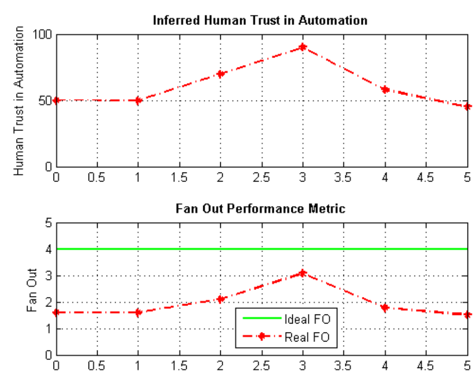
(c) Productivity Inference



(d) Human Trust in Automation

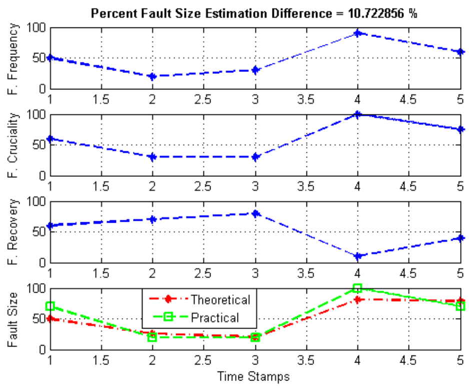


(e) Robot Attention Demand

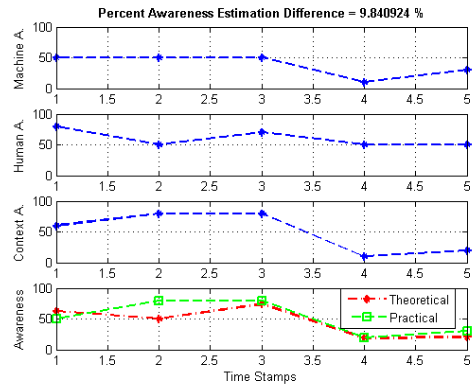


(f) Fan-out

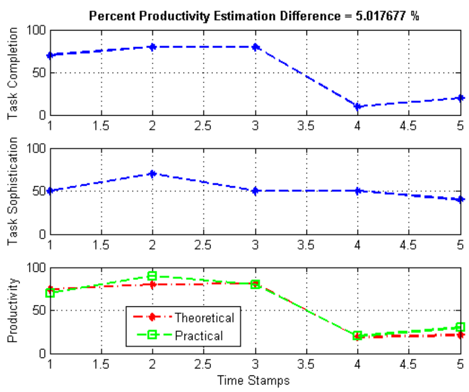
Figure 6.9: Subject #6 - Levels I and II Inferences



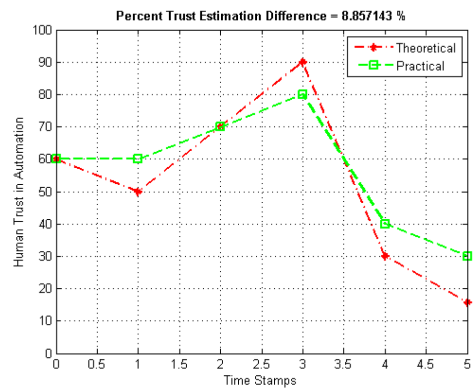
(a) Fault Size Inference



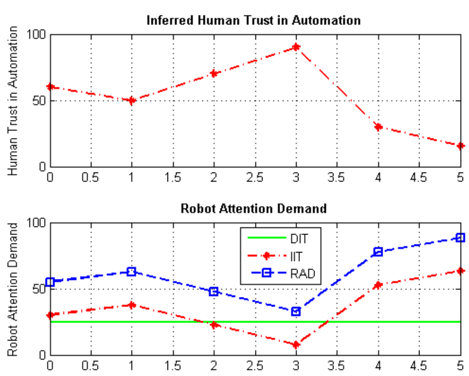
(b) Awareness Inference



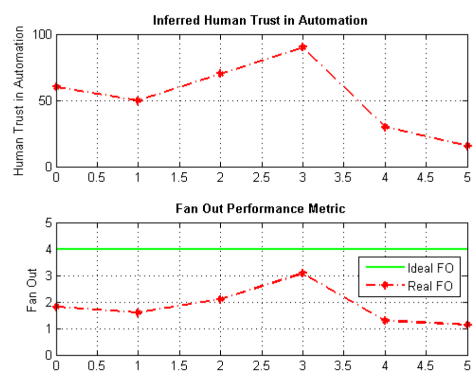
(c) Productivity Inference



(d) Human Trust in Automation

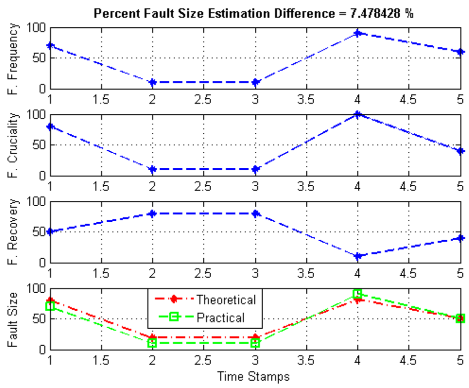


(e) Robot Attention Demand

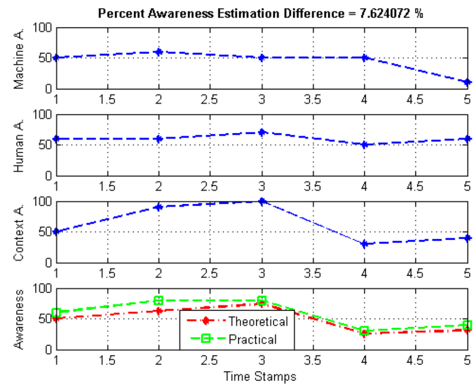


(f) Fan-out

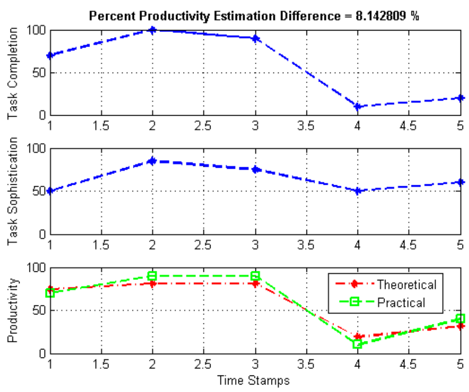
Figure 6.10: Subject #7 - Levels I and II Inferences



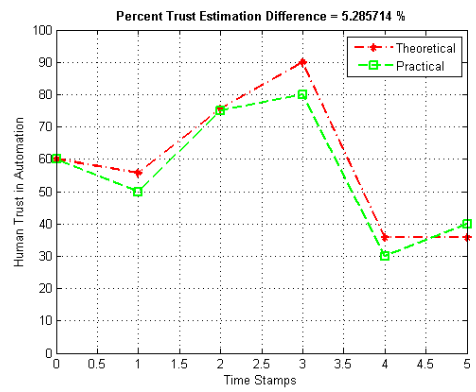
(a) Fault Size Inference



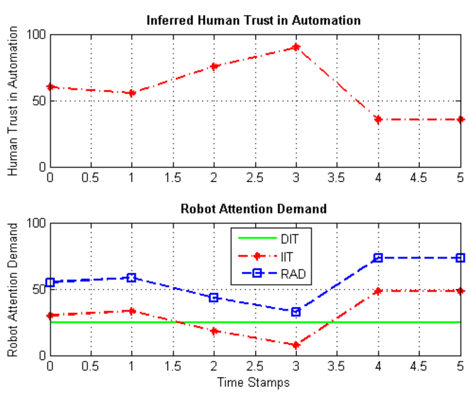
(b) Awareness Inference



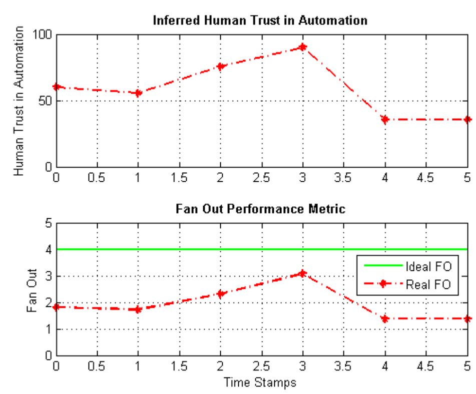
(c) Productivity Inference



(d) Human Trust in Automation

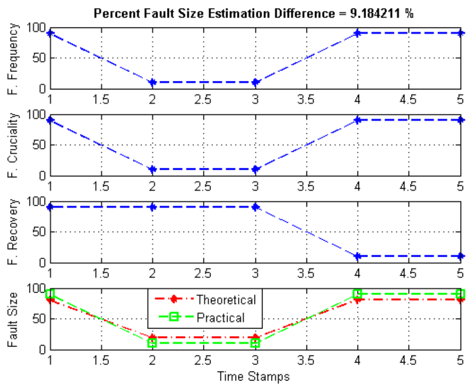


(e) Robot Attention Demand

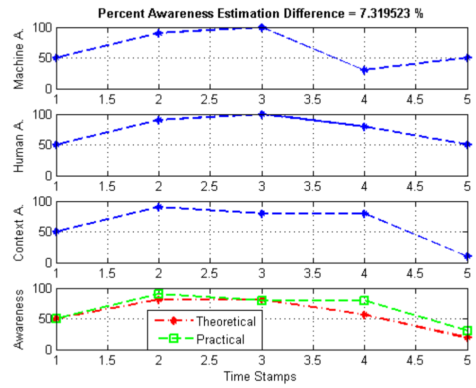


(f) Fan-out

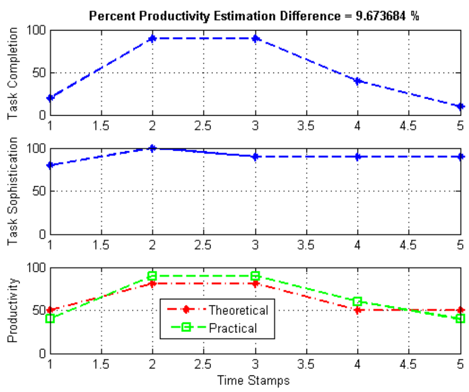
Figure 6.11: Subject #8 - Levels I and II Inferences



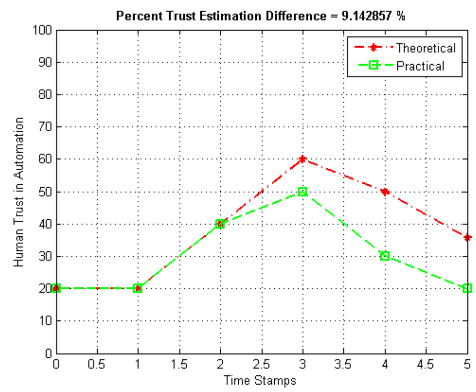
(a) Fault Size Inference



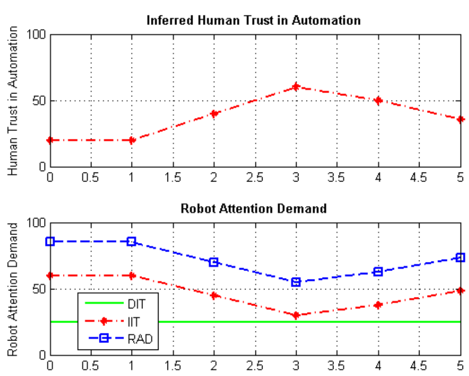
(b) Awareness Inference



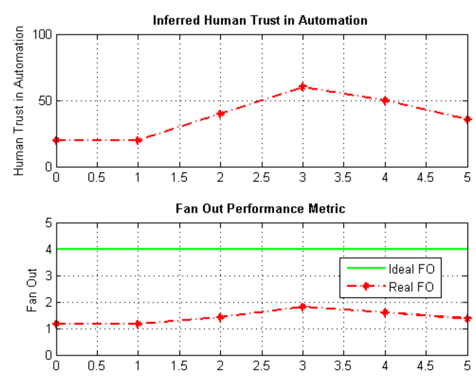
(c) Productivity Inference



(d) Human Trust in Automation



(e) Robot Attention Demand



(f) Fan-out

Figure 6.12: Subject #9 - Levels I and II Inferences

6.4 Observations and Extensions

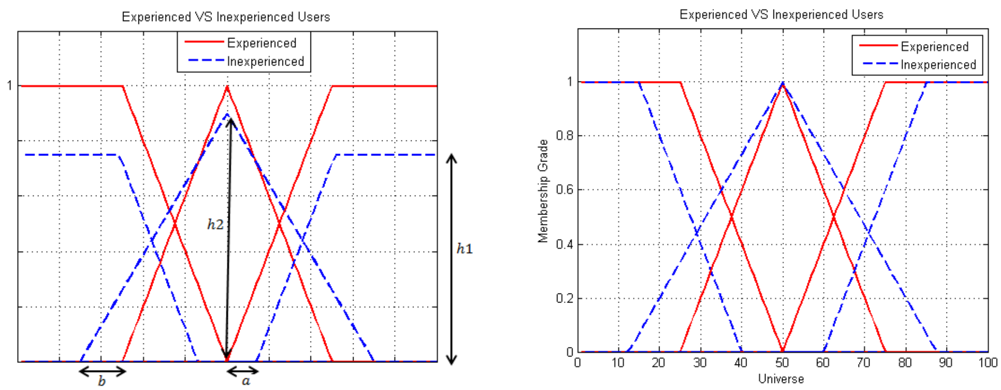
In the following, we introduce further observations and extensions to our proposed system, emphasizing their importance and relevance to this work and its future related research directions.

6.4.1 Experienced vs. Inexperienced Users

This section addresses another observation that was made when users with different expertise interacted with the robotic system. It was noted that inexperienced users who have no experience at all dealing or working with sophisticated machines or robots provided feedback that tends to show some slight differences when compared to those obtained from more experienced users in the same experimental scenarios. Inexperienced users tend to show signs of being *overimpressed* with the system when it shows successful task completion, without paying attention to minor mistakes that did not affect the overall system task completion. They also get *more frustrated* with the system when it shows strong signs of incompetence.

Toward this end, special considerations had to be taken into account to further accommodate this category of users to preserve the generic aspect of the proposed trust evaluation metric that is fed on its lowest level with first-order perceptions from the user. One solution would be to build another knowledge base to accommodate such an audience. This, however, adds further complexity to the system. This problem could be avoided with the use of *special* fuzzy sets for those inexperienced users, as shown in Figure 6.13(b). Therefore, the role of the new membership functions is to adjust the user's feedback to become more compatible with an experienced one. An ideal variation of the original membership functions is to be found. In doing so, a total of seventy-five feedbacks obtained from both experienced and inexperienced users for the same scenarios are recorded and used to optimize the support set and the height of the membership functions.

Four parameters are used in the optimization process: a, b, h1, and h2, as shown in Figure 6.13(a). The optimization process is to search for the optimal combination of values a, b, h1, and h2, so to minimize the total error between both experienced and inexperienced users' perceptions. This is achieved by reducing the overall fuzzification error between the two set of users. Values of 10, 12.5, 1, and 1, as shown in Figure 6.13(b), are found to achieve such minimal error (9.96%). Naive brute force search is used in this work as the search space is not too large and can be spanned



(a) Parameter to Learn

(b) Optimized Sets

Figure 6.13: Experienced Vs Inexperienced MFs

efficiently. Figure 6.14 shows a randomly selected sample array of a user's feedback for five scenarios. The figure shows the percent error reduction for all ''low'', ''medium'', and ''high''fuzzified membership grades. Results show significant error reduction when the new fuzzy sets are put in place.

Although this preliminary suggested solution helps reduce the fuzzification error between the two sets of users, this, however, does not solve the problem that users in general tend to provide subjective perceptions based on personal and relative judgements that vary from one person to another, thus introducing further fuzziness into the system. Such two-dimensional fuzziness can be addressed with the use of fuzzy type-2 sets. This type of fuzzy systems will be a key problem solution that we will address in our related future work.

6.4.2 Computing with Words

So far in this work, the user's first-order perceptions were received in terms of numerical values on a scale of 10. This method of expressing perceptions, however, seems less likely to occur in real environments. The user is more likely to express their perceptions using *words*, and say that the productivity was high or very high as opposed to the fact that it was 7 or 9 on a scale of 10. Humans think in relative ranges; our linguistic statements about perceptions and observations include adjectives, adverbs, intensifiers, descriptors, and/or other modifiers. This method of computing is addressed by Zadeh as computing with words (CWW) [140], in which the objects of computation are words and propositions drawn from a natural

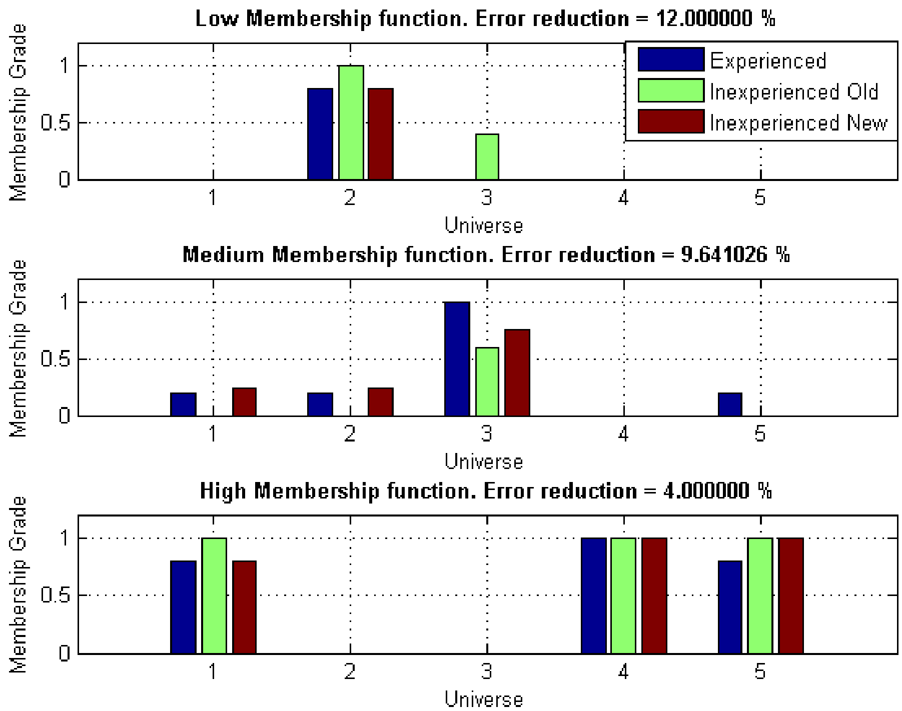
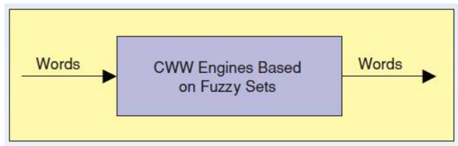


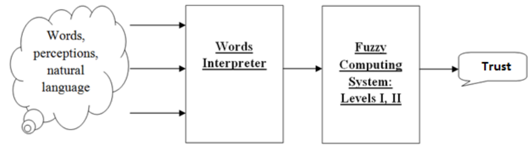
Figure 6.14: Old vs New Membership Functions - Inexperienced Users Error Reduction

language, as shown in Figure 6.15(a).

CWW is the interdisciplinary field that attempts to merge contributions from the fields of computational science, fuzzy logic, and natural language understanding. Supporting this theory, cognitive psychologist Eleanor Rosch, the mother of ''prototype theory''[141], showed in a study that involves a series of experiments that English speakers show a consistent behaviour in numerically mapping modifiers into a particular range. Such findings are often used to support the mapping of words onto values, and also to show that human cognition and categorization are based on physical human perception. Therefore, our augmented framework suggests the following: the new module added to our design is the word interpreter (WI), as shown in Figure 6.15(b). The WI accepts linguistic first-order perceptions and maps them into numerical values that are fed to level I for fuzzification. Perceptions could be: very low (ex: the fault frequency is very low), low (ex: context awareness is



(a) Computing with Words



(b) Augmented Framework

Figure 6.15: Augmented Framework - Computing with Words

low), medium (ex: fault severity is medium), high (ex: task completion is high), and very high (task sophistication is very high). Such perceptions are mapped to value 1, 3, 5, 7, 9 accordingly. Computing with words, however, is not an easy task, and a special consideration for linguistic modifiers and hedges should be carefully taken into account. A user, for instance, may choose to describe fault recovery by ''somewhat high'', or task completion by ''somewhere between low and medium''or ''not very low''. Therefore, future work will focus on careful implementation of such a module to take full advantage of the power of computing with words.

6.5 Qualitative Results Assessment

In this work, we present a further step toward identifying a common generic metric to assess the performance of the human-robot team, and the nature of the relationship that governs their behaviour while collaboratively interacting with each other to achieve some tasks. It presents a good indicator of how well the human and the robot are performing as team.

Simulations and experimental results present a practical ground which supports the abstraction and intuition of the proposed metric to efficiently assess the performance of the human-robot team, in a generic way that makes it feasible for this metric to translate well between different application domains. The metric can be used for applications ranging from navigation, to search, object detection, colour tracking, obstacle avoidance, and assistive robotics, among other applications. Results also support the proposed two-level fuzzy system architecture to model the crucial human trust in automation phenomenon, and show acceptable trust approximation and good inferences in both levels I and II, which reflects proper and representative knowledge bases design. They also show easy extension and fine-tuning of the fuzzy knowledge base to best describe a human expert's knowledge, avoiding complex mathematical models, or further offline training.

The proposed metric further allows the inference of numerical quantitative indication of both RAD and FO, hence avoiding the need to experimentally compute such values, which is often difficult, and restricted by practical limitations, such as task saturation and crowded work space, as in the case of fan-out. The metric also offers extended and generalized intuitive mathematical models to accommodate for the scenarios where multiple robots can all be a part of one bigger robotic system.

Such a metric however, faces several challenges. Although the proposed knowledge base shows good knowledge representation, such knowledge is not precise, and is subject to variation from one user to another. Human users in general tend to provide subjective perceptions based on personal and relative judgements, which vary from one person to another, thus introducing further fuzziness into the system. Furthermore, although studies show that English speakers show a consistent behaviour in numerically mapping modifiers into a particular range, the boundaries of such a range are fuzzy. These challenges, however, will be addressed by introducing another level of fuzziness into the system, to provide a fuzzy indicator (as in a range) of the system performance, rather than a single value. Values in such a range can be further given a confidence score, which can be closely related to the human expertise, level of training, and/or confidence level. This higher dimensional fuzziness shall be explored with the use of interval fuzzy type-2 systems, which can provide an indicator of a system performance with a foot of uncertainty. This will be addressed in our future research work.

6.6 Chapter Summary

In this chapter, we present a set of experiments that involve two Peoplebot robots, working collaboratively with a human operator toward achieving some tasks. The purpose of the experiments is to support the correctness and validity of the proposed fuzzy knowledge base, and tune rules where needed to best accommodate and represent the human expert's knowledge. Users were exposed to a set of five to six scenarios where the robot attempts to complete a set of different tasks, with varying levels of completion, under the command and operation of the human user. Human trust in automation, along with other first- and second-order perceptions, are marked at different time units and compared to those obtained/inferred using our proposed framework. Results show that the proposed system, with its set of modified rules, is representative and within reasonable accuracy.

This chapter also discusses further extensions of the system to accommodate for users with different expertise working with sophisticated machines. This is addressed through the use of *special* fuzzy sets, whose purpose is to minimize the total error between both experienced and inexperienced user perceptions. Finally, we address the concept of computing with words (CWW) [140], in which the objects of computation are words and propositions drawn from a natural language. Therefore, our augmented framework suggests a new module that accepts linguistic first-order perceptions and maps them to numerical values that are fed to level I for fuzzification. Future work will focus on careful implementation of such a module to take full advantage of the power of computing with words.

Chapter 7

Conclusion and Future Work

The obvious goal of any human-robot interaction system is to increase the effectiveness of the team in accomplishing some task. Therefore designing a performance metric that can assess this effectiveness is crucial. In fact, choosing meaningful performance measures provides robotics researchers with a common ground for interpretation and comparison. We believe that such evaluation criteria should focus on the human and the robot as a team.

Human-robot performance evaluation metrics have been receiving a good deal of researchers' attention, especially with the fast growth in the fields of robotics and human-robot interaction systems, and the emergence of higher-order functions, where robots are becoming more involved in increasingly more complex and less structured tasks and activities, that require indispensable interaction with people to complete the required tasks. However, much research that focuses on performance assessment of systems having both the human and the robot tends to disregard the capability of one of the agents; therefore, approaches that integrate the contributions of both the human and the robotic agents have been minimally addressed. Add to this, the lack of a generalized set of performance metrics that can span much of the robotics and HRI applications space, where most of the presented set of metrics are domain-specific or biased toward a specific application domain.

In this work, we propose a further step toward generalizing a common performance metric for assessing the human-robot team performance, by integrating both the human's and the robot's contribution in the assessment loop. Toward the efficient modelling of such metrics, we attempt to determine the true amount of time that an operator has to dedicate to the robot. Therefore, we define the robot attention demand (RAD) as a function of both direct interaction time (DIT) and indirect interaction time (IIT), where the IIT is a direct consequence of human trust in automation, which is a key factor in determining the nature of the relationship between the human user and the robot. We propose a two-level trust evaluation model which estimates the human trust in automation. This model combines the advantages of fuzzy logic and finite state machines to best model this phenomenon. The model reduces the system complexity and the size of the knowledge base by grouping perceptions into first- and second-order perceptions. Another time-based human reliability assessment model that uses a finite fuzzy state machine to estimate the human reliability state is also proposed; first-order Sugeno-like consequents are used for mapping the fuzzy states into a final crisp output. First-order consequents are used as the human reliability degrades naturally with time even when the task complexity is simple and does not impose much physical and cognitive load on the human operator. Then, generalization models that extend the proposed metric framework to accommodate multi-robot systems are proposed. Several models are derived for different task completion scenarios: sequential and parallel execution of tasks are both addressed, and with varying levels of dependency. Intuitively derived mathematical models are presented for each case. Simulations and experimental results come to support the proposed performance metric. The fuzzy knowledge bases are further updated by implementing a robotic platform where robots and users interact via semi-natural language to complete tasks with varying levels of complexity and success. User feedback is recorded and used to tune the knowledge base where needed. This comes to support the correctness and the validity of the proposed fuzzy knowledge bases, and to tune rules where needed to best accommodate and represent the human expert's knowledge.

This work presents a further step toward identifying a common performance metric for evaluating the human-robot interaction performance. It intends to provide an interaction performance index of the human-robotic system. Each interacting device is further equipped with such an interaction measure, which can be provided through an interactive performance assessment process. This process could occur offline to support different types of users, ranging from non-experienced, to semi-experienced, and fully experienced. However, much more work still needs to be carefully addressed in related future work:

- in this work, we present an insight on how to model the human reliability factor. This factor, however, requires further intensive and in depth studies in the disciplines of human sociology, physiology, psychology and so on. Therefore, careful implementation of such a module is crucial.

- in this work, we propose and implement an application robotic platform, where human users and robots interact via semi-natural language to achieve some tasks. The set of experiments involves two Peoplebot robots, working singly or together toward independent tasks, and a human operator. The tasks vary from simple to more complex. A sample of nine users was chosen for this purpose. Further work will introduce more users, interacting with other robotic agents, executing different types of tasks, to better tune the knowledge bases to best reflect the human expert's knowledge.

- in this work, we propose some special fuzzy sets to accommodate for inexperienced users after they provided feedback that tends to show some slight differences when compared to those obtained from more experienced users in the same experimental scenarios. This, however, is not the end of the story. People generally tend to have slightly different relative judgements. Therefore, fuzzy type-2 systems, which generalize type-1 fuzzy sets and systems so that more uncertainty can be handled, will be considered in our future work to address this observation.

- in this work, we address computing with words as a module that translates users' perceptions into some numerical values so they can be fuzzified. Computing with words, however, is not an easy task, and a special consideration for linguistic modifiers and hedges should be carefully taken into account. Future work will further address this module to take full advantage of the power of computing with words.

Bibliography

[1] P. Cook. Fuzzy inference systems. Human-Computer Interface Technology: Lecture Notes. Department of Computer Science, Princeton University, NJ, USA, 2007.

[2] M. Doostfatemeh and S. C. Kremer. New directions in fuzzy automata. In *International Journal of Approximate Reasoning*, pages 175–214, 2005.

[3] Mobile robots. Peoplebot. mobilerobots.com.

[4] D. Feil-Seifer and M. J. Matari. Human-robot interaction. Technical report, Center for Robotics and Embedded Systems, USC Viterbi, School of Engineering, 2008.

[5] R. R. Murphy. Human-robot interaction in rescue robotics. *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, 34:138–153, 2004.

[6] Z. Zhang and G. Nejat. Robot-assisted 3d mapping of unknown cluttered usar environments. *Advanced Robotics Journal: the Disaster Robots Special Issue*, 23:1179–1198, 2009.

[7] A. Green, H. Huttenrauch, M. Norman, L. Oestreicher, and K.S. Eklundh. User centered design for intelligent service robots. In *Proceedings of the International Workshop on Robot and Human Interactive Communication*, 2000.

[8] M.J. Mataric, J. Eriksson, D.J. Feil-Seifer, and C.J. Winstein. Socially assistive robotics for post-stroke rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 4, 2007.

[9] T. Weiner. A new model army soldier rolls closer to the battlefield. New York Times, 2005.

[10] A. Steinfeld, T. W. Fong, D. Kaber, M. Lewis, J. Scholtz, A. Schultz, and M. Goodrich. Common metrics for human-robot interaction. In *Human-Robot Interaction Conference*, 2006.

[11] J. Scholtz, J. Young, J. Drury, and H. Yanco. Evaluation of human-robot interaction awareness in search and rescue. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2004.

[12] N. Cooke, E. Salas, P. Kiekel, and B. Bell. In measuring team cognition. in Team cognition: Understanding the Factors that Drive Process and Performance, E. Salas and S.M. Fiore, Editors. American Psychological Association: Washington, D. C. p.83-106, 2004.

[13] B. Donmez, P. Pina, and M. Cummings. Evaluation criteria for human-automation performance metrics. In *PerMIS Conference Proceedings*, 2008.

[14] M. R. Endsley. Theoretical underpinnings of situation awareness. A critical review, pages 3-32. Lawrence Erlbaum Associates, Mahwah, NJ, USA, 2000.

[15] M. R. Endsley. Design and evaluation for situation awareness enhancement. *Proceedings of the Human Factors Society 32nd Annual Meeting*, 1:97–101, 1988.

[16] K. M. Tsui, H. Yanco, D. Feil-Seifer, and M. Mataric. Survey of domain-specific performance measures in assistive robotic technology. In *Proceedings of the Performance Metrics for Intelligent Systems Workshop (PerMIS)*, pages 116–123, 2008.

[17] A. Howard. A methodology to assess performance of human-robotic systems in achievement of collective tasks. In *Proceedings IEEE/RSJ International Conference Intelligent Robots and Systems*, pages 90–95, 2005.

[18] A Steinfeld, P. A. Quinones, J. Zimmerman, S. R. Bennett, and D. Siewiorek. Survey measures for evaluation of cognitive assistants. In *Proceddings NIST Performance Metrics for Intelligent Systems Workshop (PerMIS)*, 2007.

[19] M. Asada, E. Uchibe, and K. Hosoda. Cooperative behavior acquisition for mobile robots in dynamically changing real worlds via visionbased reinforcement learning and development. *Artificial Intelligence*, pages 275–292, 1999.

[20] T. Flash and T. J. Sejnowski. Computational approaches to motor control. *Current Opinion in Neurobiology*, 11:655–662, 2001.

[21] E. M. Izhikevich and G. M. Edelman. Large-scale model of mammalian thalamocortical systems. *Proceedings of the National Academy of Sciences (PNAS)*, 105:3593–3598, 2008.

[22] M. Hackel, S. Schwope, J. Fritsch, B. Wrede, and G. Sagerer. A humanoid robot platform suitable for studying embodied interaction. In *Proceedings of 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 56–61, 2005.

[23] H. Geyer, A. Seyfarth, and R. Blickhan. Compliant leg behavior explains basic dynamics of walking and running. *Proceedings of the Royal Society B: Biological Sciences*, 273:2861–2867, 2006.

[24] D. Sakamoto, T. Kanda, T. Ono, H. Ishiguro, and N. Hagita. Android as a telecommunication medium with a human like presence. In *Proceedings of 2nd ACM/IEEE International Conference on Human-Robot Interaction*, 2007.

[25] T. Takuma, S. Hayashi, and K. Hosoda. 3d bipedal robot with tunable leg compliance mechanism for multi-modal locomotion. In *Proceedings Of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.

[26] T. Minato, M. Shimada, H. Ishiguro, and S. Itakura. Development of an android robot for studying human-robot interaction. In *Proceedings of the 17th International Conference on Industrial Engineering Applications of Artificial Intelligence Expert Systems*, pages 424–434, 2004.

[27] P. Rochat and T. Striano. Perceived self in infancy. In *Infant Behavior and Development*, pages 513–530, 2000.

[28] T. Miyashita, T. Tajika, H. Ishiguro, K. Kogure, and N. Hagita. Haptic communication between humans and robots. In *Proceedings of 12th International Symposium of Robotics Research*, 2005.

[29] W. S. Harwin, R. G. Gosine, Z. Kazi, D. S. Lees, and J. L. Dallaway. A comparison of rehabilitation robotics languages and software. *Robotica*, 15:133–151, 1997.

[30] L. Robertssona, B. Iliev, R. Palma, and P. Widea. Perception modeling for human-like artificial sensor systems. *International Journal of Human-Computer Studies*, 65:446–459, 2007.

[31] T. Braun, K. Szentpetery, and K. Berns. Detecting and following humans with a mobile robot. In *Industrial Imaging and Machine Vision*, 2005.

[32] K. Kawamura, R.T. Pack, and M.Iskarous. Design philosophy for service robots. In *IEEE International Conference on Systems, Man and Cybernetics*, 1995.

[33] J. Abou Saleh, M. Baklouti, and K. Fakhri. Boosting neural networks for face detection. In *4th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management*, 2009.

[34] R. Stiefelhagen, C. Fugen, P. Gieselmann, H. Holzapfel, K. Nickel, and A. Waibel. Natural human robot interaction using speech, head pose and gestures. In *International Conference on Intelligent Robots and Systems*, 2004.

[35] M. Yoshizaki, A. Nakamura, and Y. Kuno. Vision speech system adapting to the user and environment for service robots. In *International Conference on Intelligent Robots and Systems*, 2003.

[36] M. Yoshizaki, A. Nakamura, and Y. Kuno. Mutual assistance between speech and vision for human robot interface. In *International Conference on Intelligent Robots and Systems*, 2002.

[37] O. Rogalla, M. Ehrenmann, R. Zollner, R. Becher, and R. Dillmann. Using gesture and speech control for commanding a robot assistant. In *International Workshop on Robot and Human Interactive Communication*, 2002.

[38] R. Kurnia, Md. A. Hossain, A. Nakamura, and Y. Kuno. Object recognition through human-robot interaction by speech. In *IEEE International Workshop on Robot and Human Interactive Communication*, 2004.

[39] J.Yang, W.Lu, and A.Waibel. Skin-color modeling and adaptation. School of Computer Science, Carnegie Mellon University,Pittsburgh, PA, USA. Springer Berlin, 1997.

[40] F. Karray, M. Alemzadeh, J. Abou Saleh, and M. N. Arab. Human-computer interaction: Overview on state of the art. *International Journal on Smart Sensing and Intelligent Systems*, 1:137–159, 2008.

[41] R. Bischoff and V. Graefe. Integrating vision, touch and natural language in the control of a situation-oriented behavior-based humanoid robot. In *IEEE International Conference on Systems, Man and Cybernetics*, 1999.

[42] M. W. Kadous, R. K.-M. Sheh, and C. Sammut. Caster: A robot for urban search and rescue. In *Australasian Conference on Robotics and Automation*, 2005.

[43] J. Casper and R. R. Murphy. Human-robot interactions during the robot assisted urban search and rescue response at the world trade center. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 33:367–385, 2003.

[44] I. Nourbakhsh, K. Sycara, M. Koes, M. Young, M. Lewis, and S. Burion. Human-robot teaming for search and rescue. In *IEEE Pervasive Computing*, 2005.

[45] B. Doroodgar, M. Ficocelli, B. Mobedi, and G. Nejat. The search for survivors: Cooperative human-robot interaction in search and rescue environments using semi-autonomous robots. In *2010 IEEE International Conference on Robotics and Automation*, 2010.

[46] C. Burgar, P. Lum, P. Shor, and H. Van Der Loos. Development of robots for rehabilitation therapy: The palo alto va/standford experience. *Journal of Rehabilitation Research and Development*, 37:663–673, 2002.

[47] R. Mahoney, H. Van Der Loos, P. Lum, and C. Burgar. Robotic stroke therapy assistant. *Robotica*, 21:33–44, 2003.

[48] J. Glover, D. Holstius, M. Manojlovich, K. Montgomery, A. Powers, J. Wu, S. Kiesler, J. Matthews, and S. Thrun. A robotically augmented walker for older adults. Technical report, Technical Report, Carnegie Mellon University, Computer Science Department, Pittsburgh, 2003.

[49] R. S. Rao, K. Conn, S. H. Jung, J. Katupitiya, T. Kientz, V. Kumar, J. Ostrowski, S. Patel, and C. J. Taylor. Human robot interaction: Application to smart wheelchairs. In *Proceedings of IEEE International Conference on Robotics and Automation*, 2002.

[50] A. Giminez, C. Balaguer, S. M. Sabatini, and V. Genovese. The mats robotic system to assist disabled people in their home environments. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 2003.

[51] B. Graf, M. Hans, J. Kubacki, and R. Schraft. Robotic home assistant care-o-bot ii. In *Proceedings of the Joint EMBS/BMES Confrerence*, 2002.

[52] C. Plaisant, A. Druin, C. Lathan, K. Dakhane, K. Edwards, J. Vice, and J. Montemayor. A storytelling robot for pediatric rehabilitation. In *Proceedings of the Fourth International ACM Conference on Assistive Technologies*, 2000.

[53] E. Z. F. Liu. Early adolescents perceptions of educational robots and learning of robotics. *British Journal of Educational Technology*, 41:E44–E47, 2010.

[54] K. Kang, S. Freedman, M. Mataric, M. Cunningham, and B. Lopez. A hands-off physical therapy assistance robot for cardiac patients. In *Proceedings of the International Conference on Rehabilitation Robotics*, 2005.

[55] S. Dubowsky, F. Genot, S. Godding, H. Kozono, A. Skwersky, H. Yu, and L. Shen Yu. Pamm - a robotic aid to the elderly for mobility assistance and monitoring. In *In IEEE International Conference on Robotics and Automation*, 2000.

[56] M. Montemerlo, J. Pineau, N. Roy, S. Thrun, and V. Verma. Experiences with a mobile robotic guide for the elderly. In *AAAI National Conference on Artificial Intelligence*, 2002.

[57] M. E. Pollack. Intelligent technology for an aging population: The use of ai to assist elders with cognitive impairment. *AI Magazine*, 26:9–24, 2005.

[58] J. Jung, J. Do, Y. Kim, K. Suh, D. Kim, and Z.Z. Bien. Advanced robotic residence for the elderly/the handicapped: Realization and user evaluation. In *Proceedings of the International Conference on Rehabilitation Robotics*, 2005.

[59] A. Trafton. Assistive robot adapts to people, new places. MIT News Office, 2007.

[60] W. D. Stiehl, J. Lieberman, C. Breazeal, L. Basel, L. Lalla, and M. Wolf. The design of the huggable: A therapeutic robotic companion for relational, affective touch. In *Proceedings of the AAAI Fall Symposium on Caring Machines: AI in Eldercare*, 2006.

[61] K. Wada, T. Shibata, T. Saito, K. Sakamoto, and K. Tanie. Psychological and social effects of one year robot assisted activity on elderly people at a health service facility for the aged. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005.

[62] K. Wada and T. Shibata. Living with seal robotsits sociopsychological and physiological influences on the elderly at a care house. *IEEE Transactions on Robotics*, 23:972–980, 2007.

[63] B. Scassellati. Quantitative metrics of social response for autism diagnosis. In *IEEE International Workshop on Robots and Human Interactive Communication*, 2005.

[64] B. Scassellati. Using social robots to study abnormal social development. In *Proceedings of the Fifth International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems*, 2005.

[65] H. Kozima, C. Nakagawa, and Y. Yasuda. Interactive robots for communication-care: A case study in autism therapy. In *IEEE International Workshop on Robot and Human Interactive Communication*, 2005.

[66] F. Michaud, J. F. Laplante, H. Larouche, A. Duquette, S. Caron, D. Letourneau, and P. Masson. Autonomous spherical mobile robot for child-development studies. *IEEE Transactions on Systems, Man and Cybernetics*, 35:471–480, 2005.

[67] I. Werry, K. Dautenhahn, B. Ogden, and W. Harwin. Can social interaction skills be taught by a social agent? the role of a robotics mediator in autism therapy. In *Proceedings of the Fourth International Conference on Cognitive Technology: Instruments of Mind*, 2001.

[68] K. Kosuge and Y. Hirata. Human-robot interaction. In *IEEE International Conference on Robotics and Biomimetics*, 2004.

[69] T. Takeda, Y. Hirata, and K. Kosuge. Dance partner robot cooperative motion generation with adjustable length of dance step stride based on physical interaction. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[70] P. C. Leger, A. Trebi-Ollennu, J. R. Wright, S. A. Maxwell, R. G. Bonitz, J. J. Biesiadecki, F. R. Hartman, B. K. Cooper, E. T. Baumgartner, and M. W. Maimone. Mars exploration rover surface operations: Driving spirit at gusev crater. In *IEEE International Conference on Systems, Man, and Cybernetics*, 2005.

[71] D. R. Olsen and M. A. Goodrich. Metrics for evaluating human-robot interactions. In *Performance Metrics for Intelligent Systems Workshop*, 2003.

[72] P. Pina, M. L. Cummings, J. W. Crandall, and M. D. Penna. Identifying generalizable metric classes to evaluate human-robot teams. In *Workshop at the 3rd Annual Conference on Human-Robot Interaction*, 2008.

[73] J. D. Gehrke. Evaluating situation awareness of autonomous systems. In *Proceedings on Performance Metrics for Intelligent Systems*, 2008.

[74] M. Goodrich and D. Olsen. Seven principles of efficient human robot interaction. In *Proceedings IEEE International Conference on Systems, Man and Cybernetics*, pages 3943–3948, 2003.

[75] J. Abou Saleh and F. Karray. Temporal fuzzy based modeling as applied to the class of man-machine interaction. In *IEEE International Conference on Fuzzy Systems*, 2010.

[76] J. Abou Saleh and F. Karray. Towards generalized performance metrics for human-robot interaction. In *International Conference on Autonomous and Intelligent Systems*, 2010.

[77] J. Abou Saleh and F. Karray. Towards unified performance metrics for multi-robot human interaction systems. In *International Conference on Autonomous and Intelligent Systems*, 2011.

[78] J. Abou Saleh, F. Karray, and Michael Morckos. A quantitative approach for evaluating human-robot interaction performance. In *Symposium on Advanced Intelligent Systems*, 2011.

[79] B. S. Dhillon. *Human reliability with human factors.* Elmsford, NY: Pergamon, 1986.

[80] J. E. Strutt, W. W. Loa, and K. Allsopp. Progress towards the development of a model for predicting human reliability. *Quality and Reliability Engineering International*, 14:1–14, 1998.

[81] C. A. Ntuen and E. H. Park. Predicting human reliability under workload and skill performance. *Analysis, Design, and Evaluation of Human-Machine Systems*, 10, 2008.

[82] P. A. Hancock and I. Vasmatzidis. Human occupational and performance limits under stress: the thermal environment as a prototypical example. *Ergonomics*, 41:1169–1191, 1998.

[83] R. Rau. Psycholophysiological assessment of human reliability in a simulated complex system. *Bio Psychology*, 42:287–300, 1996.

[84] B. Kirwan. *A guide to practical human reliability assessment.* London: Taylor Francis, 1994.

[85] W. G. Ireson, C. F. Coombs, and R. Y. Moss. *Handbook of reliability engineering and management.* 2nd Edition, McGraw-Hill, 1996.

[86] M. Deutsch. Trust and suspicion. *Journal of Conflict Resolution*, 2:265–279, 1958.

[87] J. B. Rotter. A new scale for the measurement of interpersonal trust. *Journal of Personality*, 35:651–665, 1967.

[88] B. Barber. The logic and limits of trust. Rutgers University Press, New Brunswick, NJ, 1983.

[89] J. K. Rempel, J. G. Holmes, and M. P. Zanna. Trust in close relationships. *Journal of Personality and Social Psychology*, 49:95–122, 1985.

[90] R. T. Golembiewski and M. McConkie. The centrality of interpersonal trust in group processes. In Theories of Group Processes, Edited by G. L. Cooper, John Wiley Sons, London, U.K., 131185, 1975.

[91] S. Jones and S. Marsh. Human-computer-human interaction:trust in cscw. *ACM SIGCHI Bulletin*, 29:36–40, 1997.

[92] J. D. Lee and N. Moray. Trust, control strategies and allocation of function in human-machine systems. *Ergonomics*, 35:1243–1270, 1992.

[93] R. Parasuraman and V. Riley. Humans and automation: Use, misuse, disuse, abuse. *Human Factors*, 39:230–253, 1997.

[94] B. M. Muir. Trust between humans and machines, and the design of decision aids. *International Journal of Man-Machine Studies*, 27:527–539, 1987.

[95] B. M. Muir. Trust in automation: Part i. theoretical issues in the study of trust and human intervention in automated systems. *Ergonomics*, 37:1905–1922, 1994.

[96] A. Freedy, E. DeVisser, G. Weltman, and N. Coeyman. Measurement of trust in human-robot collaboration. In *International Symposium on Collaborative Technologies and Systems*, pages 106–114, 2007.

[97] T. B. Sheridan. Trustworthiness of command and control systems. In *Proceedings of IFAC Man-Machine Systems*, pages 427–431, 2007.

[98] B. M Muir and N. Moray. Trust in automation: Part ii. experimental studies of trust and human intervention in a process control simulation. *Ergonomics*, 39:429–460, 1996.

[99] J. Y. Jian, A. M. Bisantz, and C. G. Drury. Foundations for an empirically determined scale of trust in automated systems. *International Journal of Cognitive Ergonomics*, 1:53–71, 2000.

[100] J. D. Lee and N. Moray. Trust, self-confidence, and operators adaptation to automation. *International Journal of Human-Computer Studies*, 40:153–184, 1994.

[101] M. T. Khasawneh, S. R. Bowling, X. Jiang, A. K. Gramopadhye, and B. J. Melloy. A model for predicting human trust in automated systems. In *Proceedings of the 8th Annual International Conference on Industrial Engineering Theory, Applications and Practice*, 2003.

[102] S. Zuboff. *In the Age of the Smart Machine: The Future of Work and Power*. Basic Books, New York, 1988.

[103] A J. Masalonis and Parasuraman. Trust as a contrast for evaluating automated aids: Past and future theory and research. In *Proceedings of the Human Factors and Ergonomic Society, 43rd Annual Meeting*, pages 184–188, 1999.

[104] N. Moray and T. Idnaagggaekri. Laboratory studies of trust between humans and machines in automated systems. *Transactions of the Institute of Measurement and Control*, 21, 1999.

[105] M. Madsen and S. Gregor. Measuring human computer trust. In *Proceedings of the 11th Australasian Conference on Information Systems*, 2000.

[106] L. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[107] F. Karray and C. De Silva. *Soft Computing and Intelligent Systems Design: Theory, Tools and Applications*. Addison Weseley, 2004.

[108] J. Jang, C. Sun, and E. Mizutani. *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice Hall, 1996.

[109] E. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7:1–13, 1975.

[110] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15:116–132, 1985.

[111] L. M. Reyneri. An introduction to fuzzy state automata. In *Proceedings of the International Work-Conference on Biological and Artificial Computation: From Neuroscience to Technology*, pages 273–283, 1997.

[112] J. N. Mordeson and D. S. Malik. *Fuzzy Automata and Languages: Theory and Applications*. Chapman and Hall, 2002.

[113] E. Roberts. Basics of automata theory. Department of Computer Science, Stanford University, CA, USA, 2005.

[114] J. E. Savage. *Models of Computation: Exploring the Power of Computing*. Addison-Wesley Publishing, 1998.

[115] S. Wolfram. *Theory and Applications of Cellular Automata*. World Scientific, 1986.

[116] C. Zaiontz. Circular automata. In *ACM-SE 14 Proceedings of the 14th annual Southeast Regional Conference*, 1976.

[117] M. Stoelinga. An introduction to probabilistic automata. *Bulletin of the European Association for Theoretical Computer Science*, 78:176–198, 2002.

[118] M. Ying. A formal model of computing with words. *IEEE Transactions on Fuzzy Systems*, 10:640–652, 2002.

[119] M. V. Lawson. Finite automata. Handbook of Networked and Embedded Control Systems. School of Mathematical and Computer Sciences, Heriot-Watt University, Scotland, 2004.

[120] E. S. Santos. Maximin automata. *Information and Control*, 12:363–377, 1968.

[121] L. A. Zadeh. Fuzzy languages and their relation to human and machine intelligence. Electronics Research Laboratory, University of California-Berkeley, 1971.

[122] M. G. Thomason and P. N. Marinos. Deterministic acceptors of regular fuzzy languages. *IEEE Transaction on Systems, Man, and Cybernnatics*, 4:228–230, 1974.

[123] B. Gaines and L. Kohout. The logic of automata. *International Journal of General Systems*, 2:191–208, 1976.

[124] C. W. Omlin, K. K. Thornber, and C. L. Giles. Fuzzy finite-state automata can be deterministically encoded into recurrent neural networks. *IEEE Transactions on Fuzzy Systems*, 6:52–75, 1998.

[125] R. Garitagoitia, J.R.G. de Mendivil, J. Echanobe, J.J. Astrain, and F. Farina. Deformed fuzzy automata for correcting imperfect strings of fuzzy symbols. *IEEE Transactions on Fuzzy Systems*, 11:299–310, 2003.

[126] C.A. Reiter. Fuzzy automata and life. *Complexity*, 7:19–29, 2002.

[127] A.K. Srivastava and S.P. Tiwari. A topology for fuzzy automata. In *AFSS International Conference on Fuzzy Systems*, pages 485–491, 2002.

[128] J. J. Astrain, J. E. Villadangos, J. R. Gonzalez de Mendivil, J.R. Garitagoitia, and F. Farina. Adjusting fuzzy automata for string similarity measuring. In *Proceedings of EUSFLAT Conference*, 2001.

[129] J. W. Danaher. Human error in atc system operations. *Human Factors*, 22:535–545, 1980.

[130] J. Finnegan, C. A. Rau, T. Rettig, and J. Weiss. Personnel errors and power plant reliability. In *Proceedings on Annual Reliability and Maintainability Symposium*, pages 290–297, 1980.

[131] K. Williams. Summary of unmanned aircraft accident/incident data: Human factors implications. Technical Report, 2004.

[132] C. Santoni, J.A. Neto, M.F. Vieira, and D. Scherer. Proposing strategies to prevent the human error in automated industrial environments. In *Proceedings of the 5th International Conference on Foundations of Augmented Cognition. Neuroergonomics and Operational Neuroscience: Held as Part of HCI International*, 2009.

[133] K. W. Lee, F. A. Tillman, and J. J. Higgins. A literature survey of the human reliability component in a man-machine system. *IEEE Transactions on Reliability*, 37, 1988.

[134] B. S. Dhillon. *Human reliability and error in medical system.* Series On Industrial and Systems Engineering - Medical, 2003.

[135] S. S. Rivera, P. A. Baziuk, and J. E. Nunezmcleod. Fuzzy uncertainties in human reliability analysis. In *Proceedings of the World Congress on Engineering*, 2011.

[136] R. Sisodia, D. B. Wolfe, and J. N. Sheth. *Firms of endearment.* Business Economics, 2007.

[137] J. Notter and M. Blair. Conflict and trust: expressions of culture and keys to organizational success. *Journal of Association Leadership, Center for Association Leadership*, 2:535–545, 2004.

[138] J. Love. Trust is so hard to earn, yet so easy to lose. eTapestry.com, 2009.

[139] Vestec. Voice enabling systems technology. evestech.com.

[140] L. A. Zadeh. From computing with numbers to computing with words - from manipulation of measurements to manipulation of perceptions. *International Journal of Applied Mathematics and Computer Science*, 12:307–324, 2002.

[141] E. H. Rosch. Natural categories. *Cognitive Pshchology*, 4:328–350, 1973.

Appendix A

Human Trust in Automation (Level II) - Knowledge Base

This appendix reports the knowledge base for level II of the human trust in automation model. Level II uses a finite fuzzy state machine to infer the trust state, which can take one or more of five states, very low, low, medium, high, and very high, based on some three essential second-order perceptions: fault size, productivity, and awareness, which are modelled using three membership functions: low, medium, and high. As such, twenty-seven rules are required to model each of the trust state. Tables A.1, A.2, A.3, and A.4 illustrate the proposed rules of the states: low, medium, high, and very high. The knowledge base that corresponds to the state *very low* was already described in table 3.1. Each table corresponds to a specific current state. Each row corresponds to a rule, which can be formulated by aggregating the input variables with an **AND** operator.

Table A.1: Current Trust State: Low

Rule	**Var1: FaultSize**	**Var2: Productivity**	**Var 3: Awareness**	**Output: Trust**
1	Low	Low	Low	Low
2	Low	Low	Medium	Medium
3	Low	Low	High	Medium
4	Low	Medium	Low	Medium
5	Low	Medium	Medium	Medium
6	Low	Medium	High	Medium
7	Low	High	Low	Medium
8	Low	High	Medium	Medium
9	Low	High	High	Medium
10	Medium	Low	Low	Low
11	Medium	Low	Medium	Low
12	Medium	Low	High	Medium
13	Medium	Medium	Low	Low
14	Medium	Medium	Medium	Low
15	Medium	Medium	High	Medium
16	Medium	High	Low	Medium
17	Medium	High	Medium	Medium
18	Medium	High	High	Medium
19	High	Low	Low	Very Low
20	High	Low	Medium	Low
21	High	Low	High	Low
22	High	Medium	Low	Low
23	High	Medium	Medium	Low
24	High	Medium	High	Medium
25	High	High	Low	Low
26	High	High	Medium	Medium
27	High	High	High	Medium

Table A.2: Current Trust State: Medium

Rule	Var1: FaultSize	Var2: Productivity	Var 3: Awareness	Output: Trust
1	Low	Low	Low	Low
2	Low	Low	Medium	Medium
3	Low	Low	High	Medium
4	Low	Medium	Low	Medium
5	Low	Medium	Medium	Medium
6	Low	Medium	High	High
7	Low	High	Low	Medium
8	Low	High	Medium	High
9	Low	High	High	High
10	Medium	Low	Low	Low
11	Medium	Low	Medium	Medium
12	Medium	Low	High	Medium
13	Medium	Medium	Low	Medium
14	Medium	Medium	Medium	Medium
15	Medium	Medium	High	Medium
16	Medium	High	Low	Medium
17	Medium	High	Medium	Medium
18	Medium	High	High	High
19	High	Low	Low	Low
20	High	Low	Medium	Low
21	High	Low	High	Medium
22	High	Medium	Low	Low
23	High	Medium	Medium	Medium
24	High	Medium	High	Medium
25	High	High	Low	Medium
26	High	High	Medium	Medium
27	High	High	High	Medium

Table A.3: Current Trust State: High

Rule	**Var1: FaultSize**	**Var2: Productivity**	**Var 3: Awareness**	**Output: Trust**
1	Low	Low	Low	Medium
2	Low	Low	Medium	Medium
3	Low	Low	High	Medium
4	Low	Medium	Low	Medium
5	Low	Medium	Medium	Medium
6	Low	Medium	High	High
7	Low	High	Low	Medium
8	Low	High	Medium	High
9	Low	High	High	Very High
10	Medium	Low	Low	Low
11	Medium	Low	Medium	Medium
12	Medium	Low	High	Medium
13	Medium	Medium	Low	Medium
14	Medium	Medium	Medium	Medium
15	Medium	Medium	High	Medium
16	Medium	High	Low	Medium
17	Medium	High	Medium	Medium
18	Medium	High	High	High
19	High	Low	Low	Low
20	High	Low	Medium	Low
21	High	Low	High	Medium
22	High	Medium	Low	Low
23	High	Medium	Medium	Medium
24	High	Medium	High	Medium
25	High	High	Low	Medium
26	High	High	Medium	Medium
27	High	High	High	Medium

Table A.4: Current Trust State: Very High

Rule	**Var1: FaultSize**	**Var2: Productivity**	**Var 3: Awareness**	**Output: Trust**
1	Low	Low	Low	Medium
2	Low	Low	Medium	Medium
3	Low	Low	High	High
4	Low	Medium	Low	Medium
5	Low	Medium	Medium	High
6	Low	Medium	High	High
7	Low	High	Low	High
8	Low	High	Medium	High
9	Low	High	High	Very High
10	Medium	Low	Low	Medium
11	Medium	Low	Medium	Medium
12	Medium	Low	High	Medium
13	Medium	Medium	Low	Medium
14	Medium	Medium	Medium	Medium
15	Medium	Medium	High	High
16	Medium	High	Low	Medium
17	Medium	High	Medium	High
18	Medium	High	High	High
19	High	Low	Low	Low
20	High	Low	Medium	Medium
21	High	Low	High	Medium
22	High	Medium	Low	Medium
23	High	Medium	Medium	Medium
24	High	Medium	High	Medium
25	High	High	Low	Medium
26	High	High	Medium	Medium
27	High	High	High	High

Appendix B

Human Trust in Automation (Level I) - Knowledge Base

This appendix reports the knowledge base for level I of the human trust in automation model. Level I comprises three Mamdani fuzzy inference models, that explain or infer the three second-order perceptions based on some first-order perceptions, and some knowledge base. Table B.1 illustrates the proposed set of rules corresponding to the fault size fuzzy inference model that takes as inputs the fault frequency, the fault cruciality, and the fault recovery. Each factor is modelled using three membership functions: low, medium, and high; as such, twenty-seven rules are required. Table B.2 illustrates the proposed rules corresponding to the awareness fuzzy inference model. The knowledge base that corresponds to the productivity fuzzy inference model was already described in table 3.2.

Table B.1: FIM1: Fault Size

Rule Number	**Var1: Fault Frequency**	**Var2: Fault Crutialty**	**Var3: Fault Recovery**	**Output: Fault Size**
1	Low	Low	Low	Low
2	Low	Low	Medium	Low
3	Low	Low	High	Low
4	Low	Medium	Low	Medium
5	Low	Medium	Medium	Medium
6	Low	Medium	High	Low
7	Low	High	Low	High
8	Low	High	Medium	Medium
9	Low	High	High	Medium
10	Medium	Low	Low	Medium
11	Medium	Low	Medium	Low
12	Medium	Low	High	Low
13	Medium	Medium	Low	High
14	Medium	Medium	Medium	Medium
15	Medium	Medium	High	Low
16	Medium	High	Low	High
17	Medium	High	Medium	High
18	Medium	High	High	Medium
19	High	Low	Low	High
20	High	Low	Medium	Medium
21	High	Low	High	Medium
22	High	Medium	Low	High
23	High	Medium	Medium	High
24	High	Medium	High	Medium
25	High	High	Low	High
26	High	High	Medium	High
27	High	High	High	High

Table B.2: FIM2: Awareness

Rule Number	**Var1: Machine Awareness**	**Var2: Human Awareness**	**Var3: Conext Awareness**	**Output: Overall Awareness**
1	Low	Low	Low	Low
2	Low	Low	Medium	Low
3	Low	Low	High	Low
4	Low	Medium	Low	Low
5	Low	Medium	Medium	Low
6	Low	Medium	High	Medium
7	Low	High	Low	Low
8	Low	High	Medium	Medium
9	Low	High	High	Medium
10	Medium	Low	Low	Low
11	Medium	Low	Medium	Low
12	Medium	Low	High	Medium
13	Medium	Medium	Low	Low
14	Medium	Medium	Medium	Medium
15	Medium	Medium	High	Medium
16	Medium	High	Low	Medium
17	Medium	High	Medium	Medium
18	Medium	High	High	High
19	High	Low	Low	Low
20	High	Low	Medium	Medium
21	High	Low	High	High
22	High	Medium	Low	Medium
23	High	Medium	Medium	High
24	High	Medium	High	High
25	High	High	Low	High
26	High	High	Medium	High
27	High	High	High	High

Appendix C

Human Reliability - Knowledge Base

This appendix reports the preliminary suggested knowledge base for the human reliability model. This model uses a finite fuzzy state machine to infer the human reliability state, which can take one or more of five states, very low, low, medium, high, and very high, based on some three inputs: number of subtasks, mental workload, and external and internal burden, which are modelled using three membership functions: low, medium, and high. As such, twenty-seven rules are required to model each of the human reliability state. Tables C.1, C.2, C.3, and C.4 illustrate the proposed rules of the states: high, medium, low, and very low. Each table corresponds to a specific current state. The knowledge base that corresponds to the state *very high* was already described in table 3.3

Table C.1: Current Human Reliability State: High

Rule Number	**Var1: Nb of Subtasks**	**Var2: Mental Workload**	**Var3: External Burden**	**Output: Human Reliability**
1	Low	Low	Low	Very High
2	Low	Low	Medium	Very High
3	Low	Low	High	High
4	Low	Medium	Low	Very High
5	Low	Medium	Medium	High
6	Low	Medium	High	High
7	Low	High	Low	High
8	Low	High	Medium	High
9	Low	High	High	Medium
10	Medium	Low	Low	Very High
11	Medium	Low	Medium	High
12	Medium	Low	High	High
13	Medium	Medium	Low	High
14	Medium	Medium	Medium	High
15	Medium	Medium	High	Medium
16	Medium	High	Low	Medium
17	Medium	High	Medium	Medium
18	Medium	High	High	Medium
19	High	Low	Low	High
20	High	Low	Medium	Medium
21	High	Low	High	Medium
22	High	Medium	Low	Medium
23	High	Medium	Medium	Medium
24	High	Medium	High	Medium
25	High	High	Low	Medium
26	High	High	Medium	Medium
27	High	High	High	Low

Table C.2: Current Human Reliability State: Medium

Rule Number	**Var1: Nb of Subtasks**	**Var2: Mental Workload**	**Var3: External Burden**	**Output: Human Reliability**
1	Low	Low	Low	High
2	Low	Low	Medium	High
3	Low	Low	High	High
4	Low	Medium	Low	High
5	Low	Medium	Medium	High
6	Low	Medium	High	Medium
7	Low	High	Low	Medium
8	Low	High	Medium	Medium
9	Low	High	High	Low
10	Medium	Low	Low	High
11	Medium	Low	Medium	High
12	Medium	Low	High	Medium
13	Medium	Medium	Low	High
14	Medium	Medium	Medium	Medium
15	Medium	Medium	High	Medium
16	Medium	High	Low	Medium
17	Medium	High	Medium	Medium
18	Medium	High	High	Low
19	High	Low	Low	High
20	High	Low	Medium	Medium
21	High	Low	High	Low
22	High	Medium	Low	Medium
23	High	Medium	Medium	Medium
24	High	Medium	High	Low
25	High	High	Low	Low
26	High	High	Medium	Low
27	High	High	High	Very Low

Table C.3: Current Human Reliability State: Low

Rule Number	**Var1: Nb of Subtasks**	**Var2: Mental Workload**	**Var3: External Burden**	**Output: Human Reliability**
1	Low	Low	Low	Medium
2	Low	Low	Medium	Medium
3	Low	Low	High	Medium
4	Low	Medium	Low	Medium
5	Low	Medium	Medium	Medium
6	Low	Medium	High	Low
7	Low	High	Low	Medium
8	Low	High	Medium	Low
9	Low	High	High	Low
10	Medium	Low	Low	Medium
11	Medium	Low	Medium	Medium
12	Medium	Low	High	Low
13	Medium	Medium	Low	Medium
14	Medium	Medium	Medium	Medium
15	Medium	Medium	High	Low
16	Medium	High	Low	Low
17	Medium	High	Medium	Low
18	Medium	High	High	Very Low
19	High	Low	Low	Medium
20	High	Low	Medium	Low
21	High	Low	High	Low
22	High	Medium	Low	Low
23	High	Medium	Medium	Low
24	High	Medium	High	Very Low
25	High	High	Low	Low
26	High	High	Medium	Very Low
27	High	High	High	Very Low

Table C.4: Current Human Reliability State: Very Low

Rule Number	**Var1: Nb of Subtasks**	**Var2: Mental Workload**	**Var3: External Burden**	**Output: Human Reliability**
1	Low	Low	Low	Medium
2	Low	Low	Medium	Medium
3	Low	Low	High	Medium
4	Low	Medium	Low	Medium
5	Low	Medium	Medium	Medium
6	Low	Medium	High	Low
7	Low	High	Low	Medium
8	Low	High	Medium	Low
9	Low	High	High	Very Low
10	Medium	Low	Low	Medium
11	Medium	Low	Medium	Medium
12	Medium	Low	High	Low
13	Medium	Medium	Low	Medium
14	Medium	Medium	Medium	Low
15	Medium	Medium	High	Low
16	Medium	High	Low	Low
17	Medium	High	Medium	Low
18	Medium	High	High	Very Low
19	High	Low	Low	Medium
20	High	Low	Medium	Low
21	High	Low	High	Low
22	High	Medium	Low	Low
23	High	Medium	Medium	Very Low
24	High	Medium	High	Very Low
25	High	High	Low	Very Low
26	High	High	Medium	Very Low
27	High	High	High	Very Low

Appendix D

Experimental Setup for User Feedback Assessment

This appendix reports further details on the experimental setup for the user feedback assessment. The set of experiments conducted in this work involves two Peoplebot robots, working singly or together toward achieving some tasks, and a human operator. A sample of nine users was chosen for this purpose, and each was exposed to a set of five to six scenarios, selected from a set of four main task categories. The purpose of these experiments is to support the correctness and the validity of the proposed fuzzy knowledge base, and tune rules where needed to best accommodate and represent the human expert's knowledge. Human trust in automation, along with other first- and second-order perceptions, are marked at different time units and compared to those obtained/inferred using our proposed framework.

Experimental Setup

Two PeopleBot robots are used in this work to perform some tasks under the command and the supervision of a human user/operator. Multiple scenarios are addressed to emphasize how the human trust in automation changes with time, and further discuss the implication of such change on the practical user free time, robot attention demand, and system fan-out. The tasks vary from simple to more complex. The robots are instructed using semi-natural speech commands. Four task categories were designed for this purpose:

- task 1: in this task, the robot is instructed to perform a series of simple tasks of moving a certain distance forward or backward, turning left or right at a certain angle, and/or controlling its gripper.

- task 2: in this task, the robot is instructed to pick an object from a certain location and place it at a goal location. In doing so, the robot has to navigate the environment while avoiding static and dynamic obstacles, looking for such an object. Then, after such an object is identified and located, the robot gets close to the object, performs a series of gripper actions, and then navigates to the goal destination. Finally, it places the object at such goal location.

- task 3: in order for the robot to build a map for its environment, it has first to wander in it, gathering sonar and laser sensor-based measurements, and then converting them into a map. In doing so, the robot must have the ability to recognize obstacles and successfully avoid them. In this task, the robot is instructed to navigate its surrounding working environment, gather sensor data, and build a map for the working environment.

- task 4: in this task, the robot is instructed to navigate the working environment searching for a predefined coloured object. Once located, the robot is instructed to track such object if subject to dynamic motion.

Feedback Form

Users' perceptions are helpful to enrich the expert's knowledge base, make sure it reflects a representative knowledge, and provides feedback on scenarios that could have been given lower attention at implementation time. All feedback was noted on a scale of 1 to 10 as shown in figure D.1. Each user is introduced to the system along with its capabilities and limitations, and then starts cooperating with the robots to achieve some tasks. First- and second-order perceptions, along with the operator's trust in the machine's automation were marked after the completion of each scenario using the below presented form.

Figure D.1: User Feedback Scale

Initial Trust in Automation

- On a scale of 1 (very low) to 10 (very high), assess your previous experience working with robots? Do you consider yourself comfortable dealing with robots and machines?
- How would you assess your trust in automation when you first put the system into action at time $t = 0$? (1 = very low, 10 = very high)

Fault Size Analysis

- Observing the robot in action attempting to complete an instructed action, how would you describe the number of mistakes made by the robot? (1 = very low, 10 = very high)
- How serious, in total, would you consider those mistakes are? (1 = not crucial, 10 = very crucial)
- How would you assess the ability of robot to recover from those mistakes and continue its effort toward completing the task without some serious interruption from the human operator? (1 = very low, 10 = very high)

As a conclusion, how would you describe the fault size made by the robot? (1 = very low, 10 = very high)

Awareness Analysis

- How would you assess the machine's awareness of its own capabilities and limitations? Was the robot aware of what it can and cannot do? (1 = very low, 10 = very high)
- Was the robot aware of the human operator and his/her availability for assistance if needed? (1 = strongly disagree, 10 = strongly agree)
- Was the machine aware of its environment and fully understood the task to be completed? (1 = strongly disagree, 10 = strongly agree)

On average, how would you assess the overall robot's awareness? (1 = very low, 10 = very high)

Productivity Analysis

- How would you assess the complexity and the utility of the task being completed? (1 = very simple, 10 = very complex/sophisticated)
- Was the robot able to achieve the final task or a good portion of the final task? (1 = strongly disagree, 10 = strongly agree)

On average, how would you assess the robot's productivity toward achieving the planned task? (1 = very low, 10 = very high)

Human Trust in Automation

Overall, how would you assess your trust in the robot's automation, and its ability to complete further tasks without the human operator's supervision? (1 = very low, 10 = very high).

Table D.1 shows the received feedback from one sample user. The user was exposed to five scenarios. Starting from an initial trust in automation of 7 at time unit $t = 0$, first- and second-order peceptions are noted for each scenario, along with the resulting evolution of the user's trust in the robot's automation.

Table D.1: Sample User Feedback

Time Unit	FF	FC	FR	Faut-Size	MA	CA	HA	Awareness	TC	TS	Productivity	Trust
$t = 1$	6	7	8	6	5	9	2	5	7	3	6	5
$t = 2$	3	5	8	4	6	7	7	7	8	8	8	7
$t = 3$	5	5	6	5	7	7	8	7	8	3	5	6
$t = 4$	9	9	2	9	3	8	2	4	2	7	3	4
$t = 5$	7	7	4	7	3	6	5	4	6	8	6	5